



Graduado en Ingeniería Informática

Universidad a Distancia de Madrid

Departamento de Ingeniería Informática

TRABAJO DE FIN DE GRADO

**Diseño e implementación de aplicaciones software para el aprendizaje
de Semáforos y Rendimientos en la asignatura Sistemas Operativos.**

Autor: Javier José García García

Directora: Sonia Pamplona Roche

MADRID, JUNIO DE 2020

Declaración de originalidad:

El autor de este trabajo, Javier José García García, con D.N.I. 31.265.993-T, declara que el contenido de este trabajo de fin de grado es original, y en el caso de haber utilizado las ideas o contenidos de otros trabajos de otros autores están convenientemente citados, de acuerdo con las normas de citación establecidas.

El número de palabras de este trabajo, excluyendo los anexos es: 10.842.

AGRADECIMIENTOS

El haber llegado hasta aquí ha sido fruto de una cabezonería impropia de mi hasta no hace mucho. Sin embargo el haber logrado terminar mis estudios de Grado es “culpa” de mucha gente que me ha apoyado y animado en todo momento.

En primer lugar, tengo que agradecer a mi mujer Montse y mis hijas que no han dejado de empujar cada día para que siguiera adelante. Sin vosotras habría sido imposible. Siento haber estado tan liado, ya me habéis recuperado al fin.

A mis compañeros, y sin embargo amigos, que han cedido parte de su tiempo en probar mis locuras y ayudarme a lograr algo decente que entregar, he de agradecerles su generosidad y compañerismo: Vicente, Alicia, Carlos, Fernando, Igor, Cristian, Manuel, Gustavo, Esteban, María, Antonio, Cintia y Adolfo, mil gracias porque sin vosotros no habría sido posible.

También tengo que agradecer a los profesores de la UDIMA su dedicación, interés y buen hacer. No son solo profesores, sino que se desviven por sus alumnos. No quiero menospreciar a ninguno, porque de todos he aprendido mucho, pero hay un grupo de ellos que debo mencionar por la indudable huella que han dejado en mi formación: Juan Alfonso Lara, David Lizcano, Javier Bravo o David de la Peña son miembros destacados de un colectivo fantástico.

Y por último, aunque por supuesto no menos importante, debo agradecer a Sonia Pamplona por tantas cosas que no hay papel suficiente para escribirlas, así que lo resumiré en un “gracias por todo”.

ÍNDICE

Índice	4
Resumen:	6
Palabras clave:	6
Abstract:	7
Keywords:.....	7
Capítulo 1: Introducción	8
1.1. Motivación.....	8
1.2. Objetivos.....	9
Capítulo 2: Trabajos Relacionados	11
2.1. Trabajos Relacionados.....	11
Capítulo 3: Planificación temporal y costes	13
3.1. Planificación Temporal.....	13
3.2. Costes	15
Capítulo 4: Desarrollo	18
4.1. Consideraciones Iniciales.	18
4.2. Análisis de Áreas de Interés.....	19
4.3. Propuestas Iniciales.	20
4.3.1.Propuesta semáforos.	21
4.3.2. Propuesta rendimientos.....	22
4.4. Requisitos y Decisiones de Diseño.....	24
4.4.1. Estilo de desarrollo.	24
4.4.2. Diseño Gráfico.....	24
4.5. Maqueta Inicial y Primer Test de Usabilidad.	26
4.5.1. Técnicas de Prototipado.....	26
4.5.2. Primer Test de Usabilidad.....	28
4.5.3. Resultados del Primer Test de Usabilidad.	31
4.6. Requisitos y Decisiones de Desarrollo.	34
4.6.1. Metodología.....	34

4.6.2. Lenguajes de Programación.....	34
4.6.3. Otras decisiones.....	36
4.7. Rediseño.	37
4.8. Implementación.	38
4.8.1. Estructura de la Plataforma.....	39
4.8.2. Base de datos.	39
4.8.3. Control de Usuarios y de sesión.	41
4.8.4. Controlador y Modelos de datos.....	42
4.8.5. Implementación de la plataforma.	43
4.8.6. Las páginas estáticas de las aplicaciones.....	44
4.8.7. Los Juegos y el JavaScript.....	44
4.8.8. El resultado final.....	46
4.8.9. Compatibilidad.	46
Capítulo 5: Evaluación	49
5.1. Evaluación por expertos.	49
5.2. Segundo Test de Usabilidad.....	50
5.3. Ajustes finales.....	52
Capítulo 6: Conclusiones.....	53
6.1. Conclusiones en relación a los objetivos.....	53
6.2. Trabajos futuros propuestos.....	55
Bibliografía.....	57
Índice de figuras	59
Índice de tablas	60
Anexos.....	61

RESUMEN:

Este trabajo aborda el desarrollo de dos aplicaciones formativas, centradas en los conceptos de semáforos y rendimientos, y planteadas mediante la teoría de la gamificación o ludificación que permiten una alta tasa de aprendizaje con un elevado recuerdo en el tiempo de los conceptos aprendidos.

Se ha planteado como una plataforma de aplicaciones formativas desarrollada en PHP, HTML, CSS y JavaScript y creada mediante el modelo MVC (Model View Controller) para separar la lógica de control, los datos y la interfaz de usuario en tres componentes distintos.

El diseño del entorno está basado en la apariencia del sistema Moodle de la UDIMA para facilitar la curva de aprendizaje del sistema y el ciclo de vida de desarrollo se ha realizado mediante técnicas de prototipado evolutivo que garanticen un interfaz de usuario eficaz, eficiente y que proporcione una alta satisfacción de uso.

PALABRAS CLAVE:

Semáforos, rendimientos, gamificación, usabilidad, eficiencia

ABSTRACT:

This work tackles the development of two training applications, focused on the concepts of semaphores and performance, and based on the gamification theory, which allows a high learning rate with a memory lasting effect of the concepts learned.

It has been designed as a platform for training applications developed in PHP, HTML, CSS and JavaScript and created using the MVC (Model View Controller) model to maintain separated logic control, data and user interface into three different components.

The interface design is based on the appearance of the UDIMA's Moodle platform to simplify the system's learning curve and the development cycle has been carried out using evolutionary prototyping techniques to ensure an effective and efficient user interface that provides high user satisfaction.

KEYWORDS:

Semaphores, performance, gamification, usability, efficiency

CAPÍTULO 1: INTRODUCCIÓN

1.1. Motivación

El estudio de la asignatura de Sistemas Operativos requiere de herramientas específicas para que los estudiantes descubran los “vericuetos” de la misma de una forma amena y didáctica ya que la asignatura en sí no sólo es parte fundamental del temario de Ingeniería Informática, tal y como se menciona en [1] sino que es una asignatura compleja de entender y asimilar. Tanto es así, que algunos conceptos son difíciles de trasladar al alumno sin un apoyo formativo adecuado que, en ocasiones, no resulta sencillo de diseñar o preparar, sobre todo para impartirse en enseñanzas online.

En éste sentido, la lectura del mencionado artículo [1] nos lleva a intuir que, pese a que diversos estudios o artículos, como [2]–[5], demuestran que los juegos son metodologías formativas de probada eficacia como complemento para ayudar al alumno a asimilar e interiorizar conceptos complejos o manejar con soltura técnicas intrincadas, no existen apenas trabajos basados en ellos que sirvan de apoyo para la asignatura de Sistemas Operativos. De hecho, es algo que queda de manifiesto claramente tras la lectura del mismo: hay muchos artículos y estudios sobre complementos formativos para el estudio de ésta materia pero casi ninguno apoyado en juegos.

La lectura, asimismo, de diversos Trabajos de Fin de Grado de estudiantes del Grado de Ingeniería Informática de la Udim, como pueden ser [6]–[8], centradas igualmente en la asignatura de Sistemas Operativos, muestra una marcada tendencia al uso de aplicaciones o juegos formativos orientados a facilitar el aprendizaje de ciertas áreas de la asignatura.

Además, y como se desprende de lo expuesto en ésta introducción, la existencia de una amplia variedad de metodologías y técnicas formativas complementarias a las

tradicionales, facilita la asimilación de conceptos a los estudiantes haciéndoles más amena la labor de aprendizaje y ayudándoles a fijar conceptos importantes con menos esfuerzo. También dota al profesorado de un abanico de alternativas que le permitan mezclar las metodologías tradicionales con técnicas diferentes que eviten el cansancio y el hastío a los estudiantes por realizar tareas repetitivas similares de forma constante.

1.2. Objetivos

Partiendo de las bases mencionadas en el apartado 1.1, éste Trabajo de Fin de Grado pretende lograr dos objetivos principalmente:

Por un lado, se trata de identificar las áreas de conocimiento asociadas a la asignatura de Sistemas Operativos que presentan mayor complejidad para comprender sus conceptos o entender las metodologías de trabajo y que necesitan de herramientas de apoyo adecuadas.

Por el otro, se centrará en la búsqueda de metodologías sencillas, principalmente apoyadas en técnicas de juego, que proporcionen un entorno con un alto grado de usabilidad.

En todo caso, la consecución de los mencionados objetivos se pretende alcanzar mediante dos pilares fundamentales que garantizan la fortaleza y fiabilidad del trabajo realizado:

- Un desarrollo mediante técnicas de prototipado evolutivo, tal y como se detalla en la asignatura de Interacción Persona Ordenador [9], que permita desarrollar un interfaz de usuario eficaz, eficiente y que proporcione una alta satisfacción de uso.

- El uso de estándares tanto en las herramientas como en los lenguajes de programación que garanticen la máxima calidad y compatibilidad del producto final.

Sin embargo, el planteamiento de éste Trabajo de Fin de Grado como un reto de límites personales, ha llevado a fijar una serie de objetivos secundarios que deberían tenerse en cuenta:

Principalmente, se pretende utilizar éste trabajo para aprender nuevas técnicas y utilizar elementos con los que no se está familiarizado. Esto permitirá que el trabajo sirva también para ampliar mi catálogo formativo y ampliar los horizontes profesionales en áreas en las que no estoy habituado a operar. En resumen, el TFG debe suponer un reto a la capacidad y conocimientos y no solo un trámite a superar.

Pero, además, se intentarán postular las bases para establecer una plataforma general de desarrollo y alojamiento de aplicaciones formativas, que unifique su apariencia y comportamiento y permita desarrollar en plazos muy cortos nuevas lecciones, juegos o soluciones formativas. Este objetivo puede parecer muy ambicioso, y lo es, pero su efecto y sentido se comprenderán en epígrafes posteriores.

CAPÍTULO 2: TRABAJOS RELACIONADOS

2.1. Trabajos Relacionados.

Antes de hacerse pública la lista de Trabajos de Fin de Grado propuestos por la UDIMA para este semestre, y tras haber pasado por la asignatura de Investigación Operativa, me había propuesto realizar una aplicación basada en un juego que ayudara a los estudiantes de la universidad a mejorar su estudio y aprendizaje. Dicha aspiración se vio reforzada tras la lectura del artículo [1] mencionado en la introducción en el que se observa que casi no existen trabajos representativos en éste área para la asignatura de Sistemas Operativos.

Una investigación más detallada me llevó a encontrar algunos trabajos relacionados que me han permitido asentar una base firme sobre la que sustentar el desarrollo de la solución propuesta.

El otro artículo que sirve de base al desarrollo propuesto y que terminó de afianzar la idea del desarrollo es [10] puesto que los autores son, el profesor que me inspiró la idea inicial y dos de los profesores que más me han motivado a lo largo de mi formación en el grado. Dicho artículo resultó notablemente formativo y contribuyó a enfocar algunos de los esfuerzos de desarrollo.

Sin embargo, la investigación de material relacionado me llevó a descubrir todo un movimiento que se aplica con éxito en áreas de marketing, gestión empresarial, etc y que, sin embargo, aún no se ha abierto apenas camino en el área educativa: la gamificación [11] o, como parece más correcto denominarla, la ludificación [12].

Como se menciona en la mayoría de las publicaciones consultadas, como las referenciadas en la introducción de éste trabajo, la ludificación aplicada a la educación

permite una forma de estudiar conceptos complejos mucho más sencilla y con efectos más duraderos en el tiempo, a la vez que genera aprendices más autónomos [13].

Si a todo lo mencionado hasta aquí le añadimos los Trabajos de Fin de Grado [6]–[8] de compañeros de años anteriores, tenemos la base necesaria para llevar adelante una aplicación formativa lúdica para la asignatura de Sistemas Operativos que cumpla con los objetivos expresados en el capítulo anterior.

CAPÍTULO 3: PLANIFICACIÓN TEMPORAL Y COSTES

3.1. Planificación Temporal

Como toda aplicación comercial a desarrollar de forma profesional, es importante determinar una planificación estimada de la misma antes de comenzar a trabajar, puesto que el presupuesto dependerá enormemente del tiempo que se ha de dedicar a cada uno de los apartados.

El seguimiento a lo largo del ciclo de desarrollo de la planificación inicial permitirá al profesional detectar desviaciones importantes, o no, en sus metodologías de trabajo y aprender para futuros desarrollos en qué áreas ha de mejorar sus estimaciones para optimizar los recursos y obtener productos de mejor calidad que aseguren un beneficio operativo adecuado.

El análisis de las particularidades del trabajo a desarrollar, y de los objetivos secundarios mencionado en el apartado 1.2 de éste trabajo, hace que se deba incrementar ligeramente la estimación de tiempo necesario para formación y documentación del material técnico utilizado sobre lo que podría ser un desarrollo estándar.

La planificación inicial estimada para este proyecto se ha agrupado en 5 epígrafes principales y quedaría, entonces, según lo mostrado en la Tabla 1. Cada uno de los epígrafes principales se ha subdividido en diversos epígrafes secundarios que aportan un mayor nivel de detalle de los distintos apartados en los que se dividirá el trabajo.

Por supuesto, el epígrafe más importante y costoso y que marcará el presupuesto determinado en el apartado siguiente, es el de implementación de la solución al que se

dedicará cerca del 40% del tiempo del proyecto, pero que en condiciones normales debería rondar el 50% del esfuerzo dedicado.

<i>Fase</i>	<i>Subfase</i>	<i>Horas</i>
Documentacion	Repaso materias (SSOO e IPO)	10
	Análisis Documentacion	5
	Busqueda y análisis artículos y material	20
	Documentacion Lenguajes programacion	40
Análisis Opciones y toma de decisiones	Identificación de áreas de interés	5
	Análisis de soluciones posibles	10
	Análisis de requisitos de desarrollo	5
	Análisis opciones de prototipado	5
Prototipado	Preparación prototipos	35
	Preparación test de evaluación	10
	Ejecución tests	10
	Evaluación resultados	10
Implementación	Desarrollo solución	135
	Desarrollo documentación	5
Redacción TFG	Desarrollo documento TFG	55
Total		360

Tabla 1: Planificación Inicial.

A lo largo del desarrollo se ha realizado un ejercicio de control de las horas invertidas en el mismo en cada uno de los apartados para poder analizar, de cara a las conclusiones finales, el grado de acierto en la planificación inicial. En la Tabla 2 se muestra una comparativa de la planificación inicial frente a la inversión temporal final en la que se observa que no se han producido desviaciones significativas, aunque sí pequeños desajustes en las estimaciones de Prototipado o Redacción.

Fase	Estimación Inicial	Total Invertido
Documentacion	75	72
Análisis Opciones y Toma de Decisiones	25	29
Prototipado	65	58
Implementación	140	138
Redacción TFG	55	60
TOTAL	360	357

Tabla 2: Comparativa de las Planificaciones Temporales Inicial y Real.

3.2. Costes

A la hora de establecer un presupuesto para el desarrollo hay que valorar cada uno de los epígrafes de la planificación de la Tabla 1. Obviamente no puede ni debe imputarse al cliente una formación en un cierto lenguaje o tecnología salvo que sea por necesaria imposición del trabajo o contrato, puesto que no sería ético hacerlo así.

Así, para el cálculo de un presupuesto de costes para éste proyecto, se han obviado los recursos destinados a la propia formación y solo se han tenido en cuenta los costes reales de documentación, análisis y desarrollo dedicados a la herramienta.

Por otro lado, un profesional dedicado a éste tipo de trabajos debe contar con los equipos y herramientas necesarios en su catálogo y no puede ni debe imputar un coste de equipos u herramientas con los que desarrollar su actividad. Al igual que ocurre con la formación, solo deberán imputarse este tipo de costes en caso de que se hayan determinado en el contrato o la oferta por necesidades especiales.

Por último, pero no menos importante, el coste hora del profesional o los profesionales involucrados dependerá de su nivel y experiencia y de si trabaja como autónomo o para una empresa u consultora. Por ejemplo, una consultora suele cobrar entre 20€/hora para programadores junior, hasta 90€/hora para un senior con amplia experiencia.

En éste caso, a la hora de establecer el presupuesto mostrado en la Tabla 3, parece adecuado suponer que nos encontramos ante el desarrollo de un programador freelance. En el mismo orden de cosas, establecemos que es un junior, recién graduado y que desea abrirse camino en el mercado, por lo que sus precios podrían estimarse en torno a los 20€/hora de desarrollo. No es descartable, sin embargo, que si se inicia en éste mercado deba bajar sus precios, sobre todo al principio, para hacer “clientela”.

Epígrafe	Coste/hora	Horas	Coste Total
Documentación	20,00 €	35	700,00 €
Análisis de opciones	20,00 €	25	500,00 €
Prototipado	20,00 €	65	1.300,00 €
Implementación	20,00 €	140	2.800,00 €
Documentación	20,00 €	5	100,00 €
Total			5.400,00 €
IVA		21%	1.134,00 €
Total con IVA			6.534,00 €

Tabla 3: Presupuesto estimado.

Por tanto, el presupuesto estimado de la solución se calcularía tomando como base la planificación estimada del apartado anterior y quitando de ella las horas de redacción del TFG y las horas de formación, así como suponiendo que se cobran al mismo precio todas las horas dedicadas, y sin costes extras por los desplazamientos debidos a los test con los usuarios.

Hay que reseñar que, al estar ante un desarrollo web que ha de ser alojado en un servidor para su utilización final, ha de añadirse al presupuesto el coste de alojamiento de la solución teniendo en cuenta, además, que lo normal es realizar un estudio de opciones de alojamiento con sus costes y ofrecer al cliente diversas alternativas y recomendaciones pero dejando que sea él el que determine la ubicación final. Como no es un objetivo de éste trabajo la realización de dicho estudio detallado, se ha realizado una pequeña tabla comparativa de posibles alojamientos que cumplan con los requerimientos del desarrollo, así como sus costes, y se presentan en la Tabla 4 como referencia.

Por supuesto, tratándose de un desarrollo profesional, los alojamientos seleccionados han de ser de calidad contrastada, que cumplan sobradamente con los requerimientos del proyecto y evitando alojamientos gratuitos y con limitaciones.

ISP	Tipo Alojamiento	Capacidad	Ancho banda	Coste anual
dinahosting	Hosting Profesional Linux	100 Gb	Ilimitado	118,80 €
webempresa	Plan M	5 Gb	200 Gb/mes	79,00 €
webempresa	Plan L	10 Gb	400 Gb/mes	149,00 €
interdominios	Hosting Web Ilimitado Linux	Ilimitado	Ilimitado	99,00 €
comvive	Hosting Inicio	6 Gb	20 Gb/mes	47,88 €
comvive	Hosting Profesional	12 Gb	50 Gb/mes	95,88 €
OVH	Hosting Profesional	250 Gb	ilimitado	59,88 €
arsys	Hosting Profesional	10 Gb	Ilimitado	82,80 €

Tabla 4: Comparativa de Costes de Alojamientos.

Al cliente se le podría ofertar solo un alojamiento u ofrecerle una batería de ellos junto con una descripción del tipo de servicios ofrecidos, seguridad del alojamiento, disponibilidad de la solución, etc y que él mismo decidiera su ubicación final.

Para el propósito de éste trabajo, no parece necesario realizar un estudio detallado de los proveedores de hosting disponibles pero baste con señalar que es una de las tareas habituales en éste tipo de desarrollos cuando el cliente no tiene hosting ya contratado o cuando el desarrollo propuesto demanda unas características especiales de hosting.

CAPÍTULO 4: DESARROLLO

4.1. Consideraciones Iniciales.

A día de hoy, los usuarios de las aplicaciones informáticas, tal y como se nos explica en el texto de referencia de la asignatura de Interacción Persona Ordenador [9], esperan que el manejo de una nueva aplicación sea tan intuitivo que les permita interactuar con ella sin tener que consultar extensos manuales o tener que realizar cursos de formación. Esto es así no solo por el uso de elementos gráficos ampliamente conocidos, sino por la estandarización de los interfaces y la simplificación de tareas.

Como se detallará más extensamente en el apartado de requisitos, para lograr un diseño eficiente, eficaz y que proporcione plena satisfacción a los usuarios a los que está destinado, se decidió abordar el desarrollo de las aplicaciones basándose en un prototipado evolutivo.

Sin embargo, la elección de éste tipo de desarrollo condiciona enormemente, e incluso modifica, el ciclo de vida natural del desarrollo de las aplicaciones. La estructura documental de éste capítulo se va a adaptar al ciclo de desarrollo seguido, aunque en la Figura 1 se muestra el ciclo de desarrollo completo utilizado para mejor referencia.

En éste Capítulo se abordará todo el detalle del ciclo de vida del proyecto a excepción de la fase de Evaluación (Fase 8) que se desglosará en el Capítulo 5

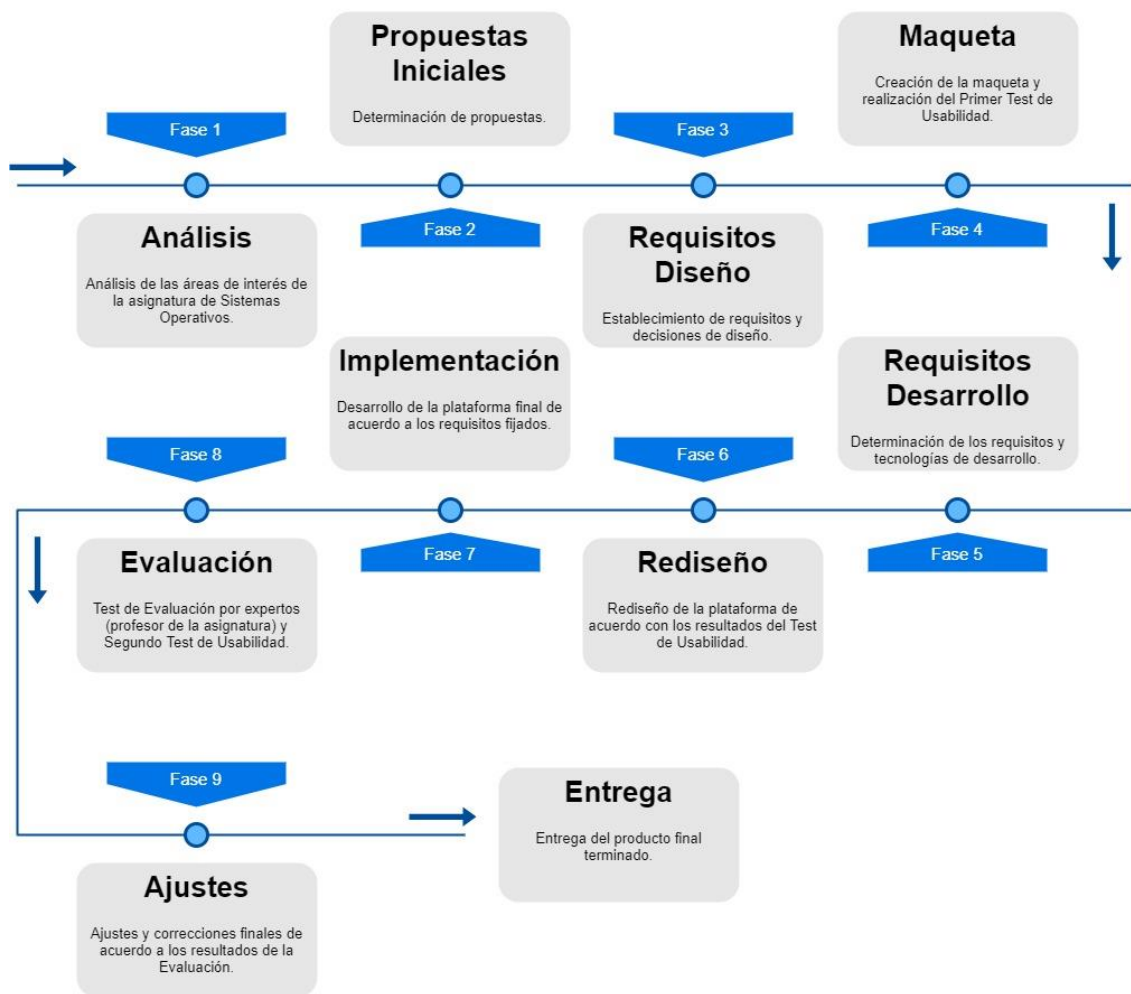


Figura 1: Ciclo de vida del proyecto.

4.2. Análisis de Áreas de Interés.

La revisión del temario de la asignatura de Sistemas Operativos impartida en la UDIMA, junto al análisis de los mencionados Trabajos de Fin de Grado de alumnos de la misma en años anteriores y ya mencionados en la motivación de éste trabajo ([6]–[8]) ha conducido a un pequeño estudio de áreas de conocimiento de la asignatura susceptibles de necesitar de herramientas de apoyo a su formación.

Por supuesto, se ha intentado evitar aquellos epígrafes o partes del temario que ya han sido abordados en anteriores trabajos y se ha buscado, por el contrario, los que pudieran suponer un reto interesante o pudieran proporcionar apoyo a conceptos complejos de abordar o entender para los alumnos de la citada asignatura.

Entre los temas no abordados anteriormente y que, al parecer, ofrecen bastante dificultad de entendimiento y aprendizaje a los alumnos de la asignatura, como se menciona en [14], los semáforos ocupan un lugar destacado. A pesar de que es un concepto, a priori, sencillo, parece que los estudiantes no logran comprender la mecánica operativa.

El otro epígrafe identificado, aunque su dificultad es relativa y su estudio más detallado se realiza posteriormente en otras asignaturas del grado, como Arquitectura de Computadores y Sistemas, es el de Problemas de Rendimiento.

Aunque podría parecer que únicamente con los semáforos se podría llevar a cabo éste trabajo, el objetivo, ya expresado pero aún no detallado, de lograr la implementación en base a una plataforma de aplicaciones, hace necesario el contar con más de un área de conocimiento sobre el que trabajar.

4.3. Propuestas Iniciales.

Una vez identificadas las áreas de conocimiento que se desea abordar en el desarrollo del TFG, hay que realizar las propuestas completas de qué y cómo se desea implementar. Tras repasar las cargas teóricas de los temas implicados, y leer artículos especializados, como [15], e incluso recursos web como [16], se llegó a la propuesta siguiente:

4.3.1.Propuesta semáforos.

La propuesta para la aplicación de semáforos es una aplicación formativa independiente compuesta de los siguientes elementos:

- Un área de teoría donde se detalle de forma sencilla y clara el funcionamiento de un semáforo para control de un recurso del Sistema Operativo. Debe estar compuesta de 2 o 3 páginas teóricas y un complemento, video o texto, que ayude a visualizar de forma muy gráfica y visual la operativa de los semáforos.
- Un juego didáctico donde prime la sencillez de manejo y que permita afianzar lo aprendido en el área teórica de una forma amena. Es necesario que la operativa del juego se plasme en un tutorial previo para que el manejo del juego sea comprendido en cualquier caso.

El código del semáforo utilizado en el apartado teórico es una ligera modificación del que se encuentra en el manual de la asignatura de Sistemas Operativos [17] pensado para hacer un poco más entendible la operativa y que aporta un elemento discordante al juego. Puede verse en la Figura 2.

Como complemento de ayuda a la teoría que haga más sencillo de comprender la misma, se ha buscado un ejemplo didáctico que explica el funcionamiento de los semáforos mediante una hipotética situación que se pretende que sea fácil de entender y en la que se identifica cada elemento del semáforo de forma clara. Se muestra en la Figura 3.

Para no alargar excesivamente este trabajo, el contenido completo de la propuesta se puede ver en el apartado 4.5 de éste capítulo dedicado a la maqueta inicial.

El código de las operaciones es:

```
wait (s){
    if (s > 0)
        s = s - 1;
        Asignar recurso al proceso;
    else
        Bloquear al proceso;
}

signal (s){
    if (!procesos bloqueados)
        s = s + 1;
    else
        Desbloquear al siguiente proceso bloqueado en la cola por la
        operación wait;
}
```

Figura 2: Código del semáforo.

Para entenderlo de una forma visual:

Imagina un taller en el que trabajan varios mecánicos pero solo hay 2 destornilladores disponibles ($s = 2$). El encargado del material dispone de un bloc de notas para tomar nota de las peticiones de uso del destornillador (*cola FIFO*).

Cuando un mecánico necesita el destornillador, pide (*wait*) al encargado que le asigne uno. Si el encargado tiene alguno disponible en el almacén ($s > 0$) se lo entrega al mecánico y reduce en 1 el stock ($s = s - 1$). Si no tiene disponible ninguno ($s \leq 0$), apunta la petición en el bloc (*añade a la cola FIFO*) y deja al mecánico a la espera de que quede libre un destornillador (*proceso bloqueado*).

Cuando un mecánico termina de usar el destornillador, lo devuelve al almacén (*signal*). El encargado comprueba en el bloc si hay mecánicos a la espera de destornilladores (*cola FIFO*). Si hay alguno, le entrega el destornillador (*desbloqueo del proceso*) y lo elimina de la lista (*cola FIFO*). Si no hay nadie a la espera, aumenta el stock en 1 ($s = s + 1$).

Figura 3: Ejemplo didáctico de semáforo

4.3.2. Propuesta rendimientos.

Al igual que la propuesta para los semáforos, la de los problemas de rendimientos consta de los mismos elementos principales, es decir, una parte teórica basada en el manual de la asignatura y un juego.

En la parte teórica se ha intentado simplificar al máximo el material y mostrar únicamente la fórmula con la que se debe realizar el cálculo y una simplificación, tipo algoritmo, de cómo debe aplicarse dicha fórmula en cada caso. Se pueden observar ambas fórmulas, tanto del cálculo del rendimiento en una planificación Round Robin, como del porcentaje de utilización de la CPU en las figuras 4 y 5.

En una planificación Round Robin el **Rendimiento** se calcula como:

$$\text{Rendimiento} = \text{número de procesos} / \text{tiempo que tardan en ejecutarse los procesos}$$

Método de cálculo:

1. Calcular las **rodajas** de cada proceso en función del tiempo por rodaja.
2. Calcular el número de **cambios de contexto** como rodajas – 1.
3. Sumar los tiempos de los dos pasos anteriores para obtener el tiempo total de proceso.
4. Calcular el **rendimiento** en μs según la fórmula anterior.

Figura 4: Rendimiento en una planificación Round Robin

El **Porcentaje de Utilización de la CPU** se calcula como:

$$\text{Porcentaje de Utilización} = \text{Tiempo que la CPU permanece ocupada} / \text{Tiempo total de ejecución}$$

Método de cálculo:

1. Calcular el tiempo de cada operación.
2. Calcular el tiempo total del programa.
3. Identificar que operaciones ocupan a la CPU.
4. Calcular el **porcentaje de utilización de la CPU** (%) según la fórmula anterior.

Figura 5: Porcentaje de utilización de la CPU

La propuesta completa se puede consultar en el apartado de la maqueta inicial, junto con la propuesta del caso de los semáforos.

4.4. Requisitos y Decisiones de Diseño.

Una vez definidas las propuestas con las que se va a trabajar y establecidos los contenidos teóricos y las operativas de los juegos, llega el momento de definir cómo se va a desarrollar el trabajo.

4.4.1. Estilo de desarrollo.

Como ya se mencionó en las consideraciones iniciales de éste capítulo, se decidió llevar a cabo el desarrollo de las aplicaciones siguiendo la metodología de prototipado evolutivo que permitiera cumplir con los atributos de usabilidad mencionados en el manual de la asignatura:

- Facilidad de aprendizaje.
- Recuerdo en el tiempo.
- Eficiencia de uso.
- Baja tasa de errores.
- Satisfacción obtenida del sistema.

Con ello se intenta obtener un diseño eficiente, eficaz y que proporcione plena satisfacción a los usuarios durante su manejo, evitando la necesidad de manuales o instrucciones complejas.

4.4.2. Diseño Gráfico.

Uno de los principales problemas que suelen afectar a éste tipo de aplicaciones es la complejidad de su entorno y la sobrecarga de elementos en pantalla. La intención de éste trabajo, desde el primer momento, es seguir la enseñanzas del manual de la

asignatura de Interacción Persona Ordenador [9] y procurar que el diseño cumpla con los parámetros de diseño visual que se detallan en la unidad 8 del mismo.

Sobre todo, era imperativo que el diseño fuera muy sencillo, claro y auto explicativo. Por dicho motivo se optó por “imitar” la apariencia visual básica del entorno Moodle de la UDIMA para que, de un primer vistazo, los usuarios de la aplicación entendieran la distribución y organización del entorno.

Además, se optó por fuentes claras, sin serif y, sobre todo en los apartados teóricos, por tamaños de fuente algo grandes con separaciones de línea mayores de la habitual que hicieran sencilla la lectura.

Se evaluaron varias soluciones de diseño para marcar en el menú en que página se encontraba el usuario de la aplicación en cada momento, y se decidió por una solución muy visual, como podrá observarse posteriormente en la maqueta, con un punto rojo junto al elemento del menú activo en ese momento.

El diseño de los juegos, sobre todo el de semáforos, fue realizado pensando en marcar diferencias entre los distintos elementos para hacer más clara su identificación de un solo vistazo y se situó en una distribución que permitiera aprovechar el espacio disponible al máximo.

Por último, pero no menos importante, se decidió limitar al máximo los elementos gráficos o imágenes estáticas en el diseño para hacerlo más ligero, así como la utilización de elementos con licencia Creative Commons, como los iconos de FontAwesome, que permiten su utilización simplemente mencionando de donde proceden y enlazando a su licencia de uso.

4.5. Maqueta Inicial y Primer Test de Usabilidad.

4.5.1. Técnicas de Prototipado.

Una vez establecidos los requisitos y las decisiones de diseño, se procedió a elaborar la primera maqueta del sistema para dar forma a los distintos elementos de las aplicaciones y al diseño elegido. De cara a simplificar el diseño y preparar las maquetas para los test de usabilidad se decidió trabajar en Powerpoint como primera opción.



Figura 6: Página 1 de Formación Teórica de la Aplicación de Semáforos

Se elaboraron varias maquetas desde el esquema inicial de distribución hasta llegar a una maqueta completa del sistema preparada para, posteriormente, poder subirlas a la plataforma de maquetado online.

En las Figuras 6 y 7 puede verse algunas de las páginas de la maqueta final con la que se preparó el primer test de usabilidad, pero el diseño completo de las maquetas puede verse con detalle en el Anexo 1.



Figura 7: Página 1 del Tutorial de la Aplicación de Semáforos

De cara a preparar dicho test se evaluaron varias soluciones de maquetado online como Invision, Sketch, Figma u otras pero se optó por la simplicidad y facilidad de Marvel para construir el prototipo en base a imágenes de alta calidad de la maqueta de PowerPoint, como puede observarse en la Figura 8.

Una vez preparada la maqueta con el diseño inicial, había que someterla al primer test de Usabilidad con usuarios potenciales reales para ver que reacciones despertaba.

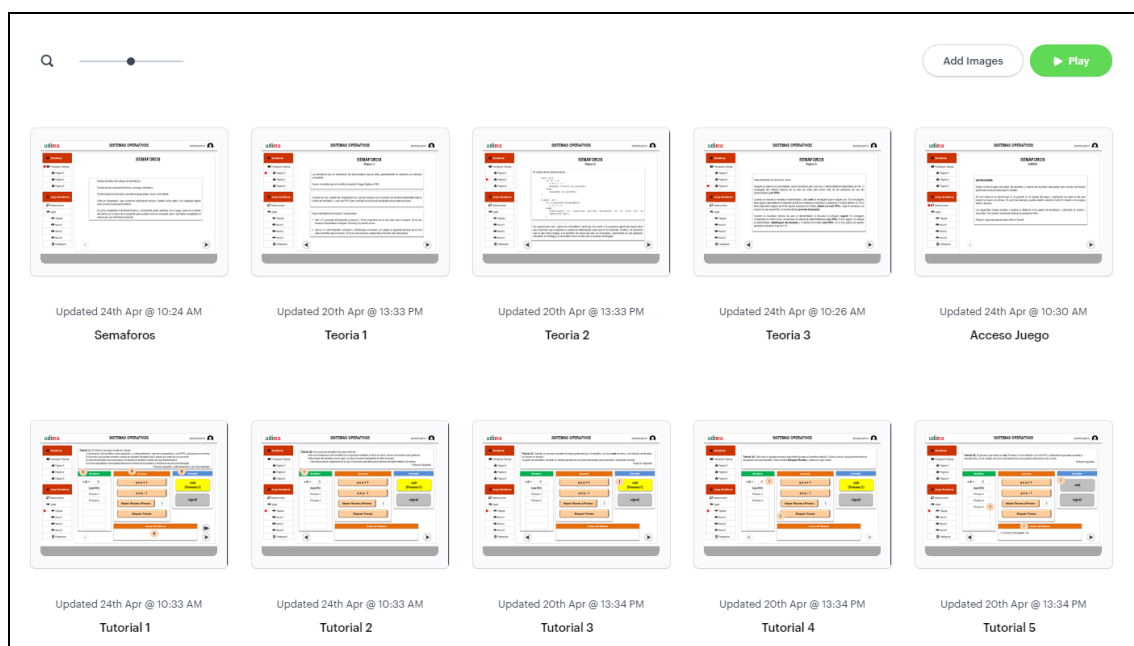


Figura 8: Detalle de la maqueta de la Aplicación de Semáforos en Marvel.

4.5.2. Primer Test de Usabilidad.

En éste punto, y dado que se pretendía completar el ciclo de desarrollo de las aplicaciones mediante el modelo de prototipado evolutivo, era necesario calibrar el tiempo disponible para poder realizar los tests de evaluación finales de la aplicación. Por dicho motivo se decidió que el primer test de usabilidad se haría asíncrono pero con un mayor número de participantes que aportaran el mayor número de valoraciones posibles.

Lo primero era redactar las instrucciones del Test de Usabilidad de forma que cumplieran con los requisitos marcados en el manual de la asignatura pero fueran lo suficientemente explicativos como para permitir que el test se realizara asíncronamente y de forma autónoma por los participantes del mismo. Asimismo, el realizar un test asíncrono hacía necesario un formulario de preguntas posterior muy detallado que

permitiera valorar de forma adecuada las reacciones que el diseño y la usabilidad de las aplicaciones habían generado en los participantes.

En las Figuras 9 y 10 puede verse un detalle parcial de ambos componentes del test pero pueden ser consultadas con detalle en el Anexo 2 al final del documento.

Instrucciones del Primer Test de Usabilidad de las Aplicaciones de Semáforos y Rendimientos para la asignatura de Sistemas Operativos.

Gracias por tu ayuda en la evaluación de los prototipos de las aplicaciones de apoyo a la formación de la asignatura de Sistemas Operativos.

En éste test se desea comprobar la usabilidad del diseño de las aplicaciones y el contenido de los apartados de Teoría e Instrucciones o Tutorial de los juegos. En ésta versión no se puede jugar aún y la pantalla del juego es únicamente para mostrar la apariencia, pulsando en cualquier punto de ella se pasa a la pantalla final del juego.

El objetivo del test es identificar problemas en la estructura del diseño de las aplicaciones y medir su grado de usabilidad por parte de los usuarios potenciales. Ten en cuenta que se trata de una maqueta gráfica y que no te encuentras en un entorno web real, por lo que las funciones que se “activan” o “desactivan” en función de si ha sido realizado antes o no (como rezan los textos explicativos) se encuentran siempre activadas dando por supuesto que ya han sido realizadas antes.

Para la realización del test debes suponer que estas accediendo a la formación online, a través de la plataforma Moodle de la Udima, de los temas de semáforos y rendimientos. Las tareas que has de realizar son:

- 1- Usa el enlace a continuación para acceder a la formación online de Semáforos:
<https://marvelapp.com/e5cfaj7>
- 2- Lee la información de la página de introducción y completa el recorrido por el módulo

Figura 9: Detalle de las Instrucciones del Primer Test de Usabilidad

Para la realización del primer test se intentó buscar la colaboración de un grupo de estudiantes de la UDIMA lo más variado posible. Por ello se pidió ayuda, a través de la profesora de la asignatura, a los estudiantes que están cursando la misma este año, obteniéndose la ayuda de 3 de ellos.

Primer Test de Usabilidad de la Apl ☆

Preguntas Respuestas 13

Sección 1 de 7

Primer Test de Usabilidad de la Aplicación de Semáforos para Sistemas Operativos

Descripción del formulario

Dirección de correo electrónico *

Dirección de correo electrónico válida

Este formulario recopila las direcciones de correo electrónico. [Cambiar configuración](#)

Después de la sección 1 Ir a la siguiente sección

Sección 2 de 7

Tareas del test

Descripción (opcional)

¿Has podido completar las tareas del test de usabilidad sin problemas? *

☐ Si

☐ Si, pero con problemas o dudas

Figura 10: Detalle del Formulario de Test a los participantes.

Por tratarse de un test asíncrono, eran necesarios al menos 10 participantes que generaran un volumen adecuado de comentarios y valoraciones. Por ello se recurrió a estudiantes de la asignatura de Sistemas Operativos y estudiantes de los últimos cursos del grado de Ingeniería informática de la UDIMA y se incorporó también a un graduado de la Universidad Carlos III que cuenta con experiencia en diseño de aplicaciones para Android.

En la Tabla 5 se muestra un resumen de los datos de los participantes en el estudio, en total 13, y sus características. Obviamente, se han omitido los nombres y los correos de los implicados por el cumplimiento del Reglamento General de Protección de Datos [18].

Participante	SSOO	Fecha	Experiencia Profesional	Universidad
Usuario1	Aprobada	30/04/2020	Alta	Udima
Usuario2	Aprobada	01/05/2020	Alta	Udima
Usuario3	Aprobada	03/05/2020	Alta	Udima
Usuario4	En curso	03/05/2020	Baja	Udima
Usuario5	Aprobada	05/05/2020	Media	Udima
Usuario6	Aprobada	05/05/2020	Media	Carlos III
Usuario7	En curso	07/05/2020	Baja	Udima
Usuario8	Aprobada	07/05/2020	Alta	Udima
Usuario9	Aprobada	07/05/2020	Alta	Udima
Usuario10	Aprobada	08/05/2020	Alta	Udima
Usuario11	Aprobada	08/05/2020	Alta	Udima
Usuario12	Aprobada	09/05/2020	Media	Udima
Usuario13	En curso	17/05/2020	Baja	Udima

Tabla 5: Participantes en el Primer Test de Usabilidad

4.5.3. Resultados del Primer Test de Usabilidad.

La buena disposición de los participantes en el test hizo que el mismo transcurriera sin incidencias. Los comentarios aportados permitieron un análisis muy completo de las aplicaciones y arrojaron luz sobre sus carencias, principalmente en el diseño, que llevaron, como veremos posteriormente, a un rediseño de la parte lúdica de ambas aplicaciones.

Aunque los detalles de los formularios se pueden consultar completos en el Anexo 3, se presentarán aquí, a continuación, algunos resúmenes de las valoraciones recibidas en la realización del Test.

En general todos los usuarios pudieron completar ambos tests sin problemas reseñables salvo alguna duda menor con los botones de siguiente o saltar, pero nada destacable, como puede verse en la Figura 11.

En cuanto a la dificultad de ambos tests, la gran mayoría los encontró fáciles o muy fáciles, con escasas divergencias.

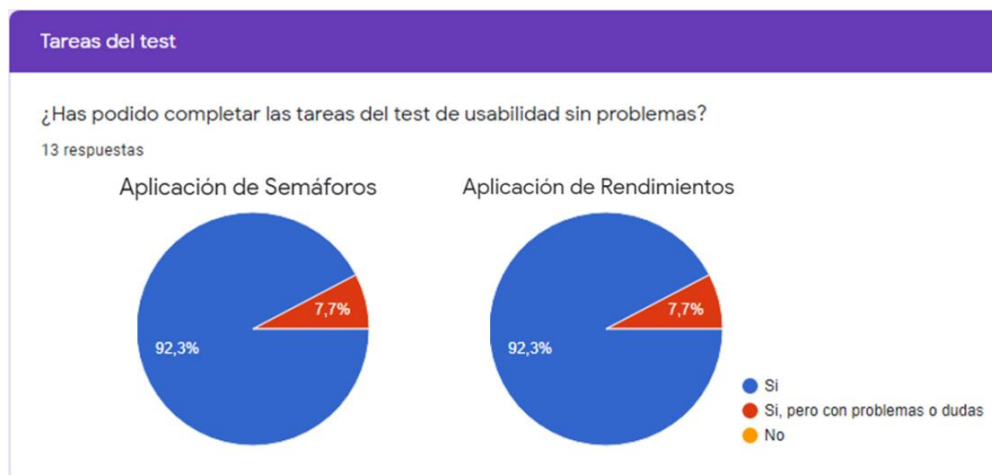


Figura 11: Valoración del Primer Test de Usabilidad: Completo.

El diseño base de las aplicaciones, así como su claridad y limpieza y funcionalidad fueron muy destacados, obteniéndose en todos los casos puntuaciones favorables, como muestran los gráficos de la Figura 12. Sin embargo, algunos usuarios apuntaron a que el uso de un punto rojo para indicar la situación en el menú no quedaba bien integrado. Igualmente, los contenidos teóricos parecieron correctos, entendibles, no sobrecargados y muy adecuados.

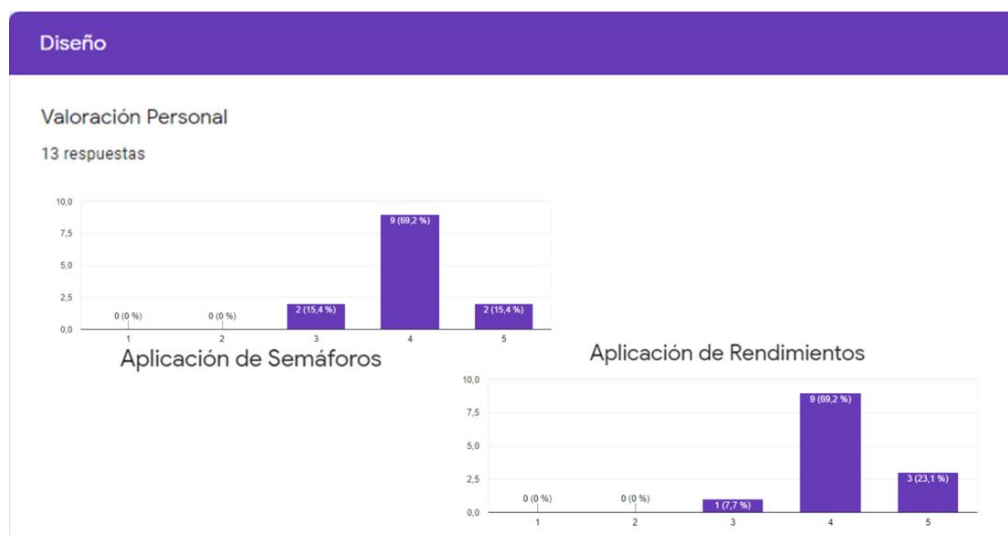


Figura 12: Valoración del Primer Test de Usabilidad: Diseño.

Sin embargo, el diseño de los juegos, sobre todo el de semáforos, no fue tan bien valorado ya que, aunque sus puntuaciones no fueron bajas, los comentarios de los participantes apuntaban a otros resultados. La mayoría de los comentarios recibidos, respecto a la aplicación de semáforos, indicaban:

- Diseños antiguos y faltos de integración. Necesidad de diseños más actuales.
- Paleta de colores no adecuada.
- Necesidad de reorganización de los elementos.
- Falta de claridad en los estados del juego.
- Necesidad de una ayuda en el juego.

Mientras que en la aplicación de rendimientos:

- Necesidad de un diseño más moderno.
- Orientar más los ejercicios a tipo test.
- No ofrecer premios económicos como objetivo del juego sino de otro tipo.
- Ejercicios separados en 3 partes para hacerlos más claros.

Además, algunos de los participantes incluso sugirió, en conversación posterior al test, que la consulta de web de Material Design de Google [19] aportaría ideas de diseño útiles tanto en el ámbito de la organización de elementos en pantalla, como en la elección de las paletas de colores adecuadas para su integración.

Por último, se reseñaron algunos fallos de redacción, errores ortográficos o gramaticales y comentarios de mejora que, por supuesto, se deben tener en cuenta para elevar la calidad global del producto final.

4.6. Requisitos y Decisiones de Desarrollo.

Antes comenzar con el rediseño de la aplicación y su implementación de acuerdo a las aportaciones recibidas en el test de usabilidad, es interesante detallar los requisitos y decisiones de desarrollo tomados así como la elección de los lenguajes de programación sobre los que se sustentará el trabajo.

4.6.1. Metodología.

Como ya se mencionó en el apartado 1.2, un objetivo de éste trabajo es desarrollar la implementación final en base a una plataforma de aplicaciones formativas y no como aplicaciones totalmente independientes. El motivo de éste tipo de desarrollo es que, aunque requieren de un mayor esfuerzo inicial de desarrollo y diseño del entorno, a la larga permiten implementar nuevos elementos de forma rápida y sencilla así como permiten alterar el diseño completo sin tener que editar un gran número de páginas distintas.

Para ello era necesario apoyarse en frameworks o librerías que garantizaran y simplificaran el núcleo de la aplicación, como por ejemplo los accesos a la necesaria base de datos, sin tener que malgastar recursos en gestionar o securizar dichos accesos. En el siguiente apartado se detallaran las decisiones tomadas en éste área.

4.6.2. Lenguajes de Programación.

Uno de los primeros retos a enfrentar es la decisión del lenguaje de programación en que se basará el desarrollo. Como es lógico, al tratarse de una aplicación web, la mayor parte de la misma ha de estar escrita en HTML con el lógico apoyo en CSS, pero el acceso a elementos de la base de datos hacía necesario el uso de un lenguaje más apropiado. Esto, unido a la idea de utilizar un framework que apoyara ciertos elementos

del desarrollo, hizo necesario realizar un pequeño estudio de opciones basado, no solo en las capacidades del lenguaje, sino también en los propios conocimientos previos que sirvieran de sustento y evitaran un innecesario desperdicio de tiempo en el aprendizaje previo.

En paralelo se inició la búsqueda de frameworks de desarrollo que aportaran los elementos necesarios, aseguraran la fiabilidad y, sobre todo, fueran lo más estándares y sencillos posible.

Al final, el conocimiento previo y la simplicidad de PHP se impusieron y la búsqueda del framework se orientó a este lenguaje. Se valoraron varios, con especial atención en Laravel, Symphony y CodeIgniter y la eficacia y facilidad de uso de éste último se impusieron frente a la potencia de sus adversarios.

CodeIgniter [20] es un framework de desarrollo web de código libre programado y mantenido por la Codeigniter Foundation. Es muy ligero y sencillo ya que, con un peso de 1.2 Mb en su versión 4.0.3 permite, mediante el uso de modelos de datos y controladores, desarrollar aplicaciones web en PHP manteniendo una alta seguridad de acceso a la base de datos. Permite llevar a cabo un desarrollo Model View Controller (o Modelo Vista Controlador) que separa los datos de la aplicación, de la interfaz de usuario y de la lógica de control e incluye elementos básicos ya preparados que simplifican el trabajo enormemente, como el control de sesiones, la protección contra ataques XSS o la protección de la base de datos contra técnicas de SQL Injection.

Su configuración y puesta en marcha son muy sencillas y permitió empezar el desarrollo en un mínimo lapso de tiempo por su facilidad para hacerlo funcionar en modo Desarrollador en cualquier sistema operativo. Por supuesto, incluye una extensa y detallada documentación que reduce el tiempo necesario para formarse en su manejo.

Por último, es necesario incluir en este apartado que, como ya se enunció en los objetivos, se pretende utilizar éste desarrollo para aprender y mejorar en el uso de técnicas y lenguajes no usados, o usados muy someramente, por el autor del mismo. Por ello, se decidió que el desarrollo de la parte operativa de los juegos se realizaría en JavaScript y, llegados a éste punto, parecía obligado probar el uso de la biblioteca jQuery para enviar los datos del juego a la base de datos. Esta decisión supone un doble reto puesto que habría que integrar jQuery con el modelo de datos de CodeIgniter.

4.6.3. Otras decisiones.

Algunas otras decisiones menores, pero que afectan a la plataforma, la implementación u otros aspectos, se fueron tomando a lo largo de las diversas etapas de desarrollo y redacción de este trabajo. Se reúnen aquí por simplicidad y para evitar que se encuentren desperdigadas por el texto.

Por un lado, la idea base para estas aplicaciones parte de un juego sencillo que el profesor David de la Peña tiene incorporado en la asignatura de Investigación Operativa. A la hora de desarrollar éste trabajo, se decidió imitar la sencillez en la implementación del tratamiento de usuarios que se hacía en la aplicación del profesor de la Peña, puesto que el usuario se pasa como parámetro desde Moodle en la llamada al juego y se mantiene en uso durante la sesión.

Así, se optó por llevar adelante el mismo tipo de implementación y gestionar el usuario únicamente por la llamada inicial y controlarlo mediante la sesión. La diferencia respecto al planteamiento del profesor de la Peña en la implementación de la plataforma parte de que si, por el motivo que sea, no se pasa un usuario en la llamada inicial a alguna de las aplicaciones, el sistema le asigna un usuario anónimo aleatorio basado en la sesión.

Por otro lado, la implementación de las aplicaciones como una plataforma y el uso de una base de datos para gestionarla, llevó a plantear, y finalmente decidir, que la mayor parte del contenido, básicamente todo lo estático, finalmente se almacenara en la misma base de datos, reduciendo a su mínima expresión el número de páginas directas almacenadas en el servidor web. Esto obliga a realizar el diseño e implementación de una base de datos en MySQL/MariaDB, que se detallará en el apartado correspondiente.

4.7. Rediseño.

Gracias a la ayuda de las personas que cedieron su tiempo desinteresadamente para evaluar las aplicaciones y a sus acertados comentarios e ideas, se acometió el proceso de rediseño de la aplicación con el apoyo, una vez más, de la aplicación Powerpoint como sustento gráfico básico para la implementación.

Se eliminaron los puntos rojos para marcar la posición en el menú y se sustituyeron por texto en negrita. Se corrigieron los errores ortográficos y gramaticales y se mejoraron los textos pero, sobre todo, se acometió una profunda reestructuración de toda la parte lúdica de las aplicaciones.

Para ello fue básico el análisis de las recomendaciones de diseño de la web de Google mencionada y el estudio de situación de los elementos. Se revisaron y analizaron las paletas de colores preparadas que ofrece dicha web y se estudió la compatibilidad y correlación de los colores globales del sistema.

Adicionalmente, se revisó la mecánica del juego y se simplificaron ciertas tareas para hacerlo más comprensible y sencillo de manejo a la par que integrado visualmente mejor en el nuevo diseño.

Con todo ello, se obtuvo un diseño listo para su paso a la plataforma de implementación, como se muestra en la Figura 13. Todo el diseño está construido mediante elementos HTML y CSS y las únicas imágenes utilizadas son los leds de colores de las llamadas.

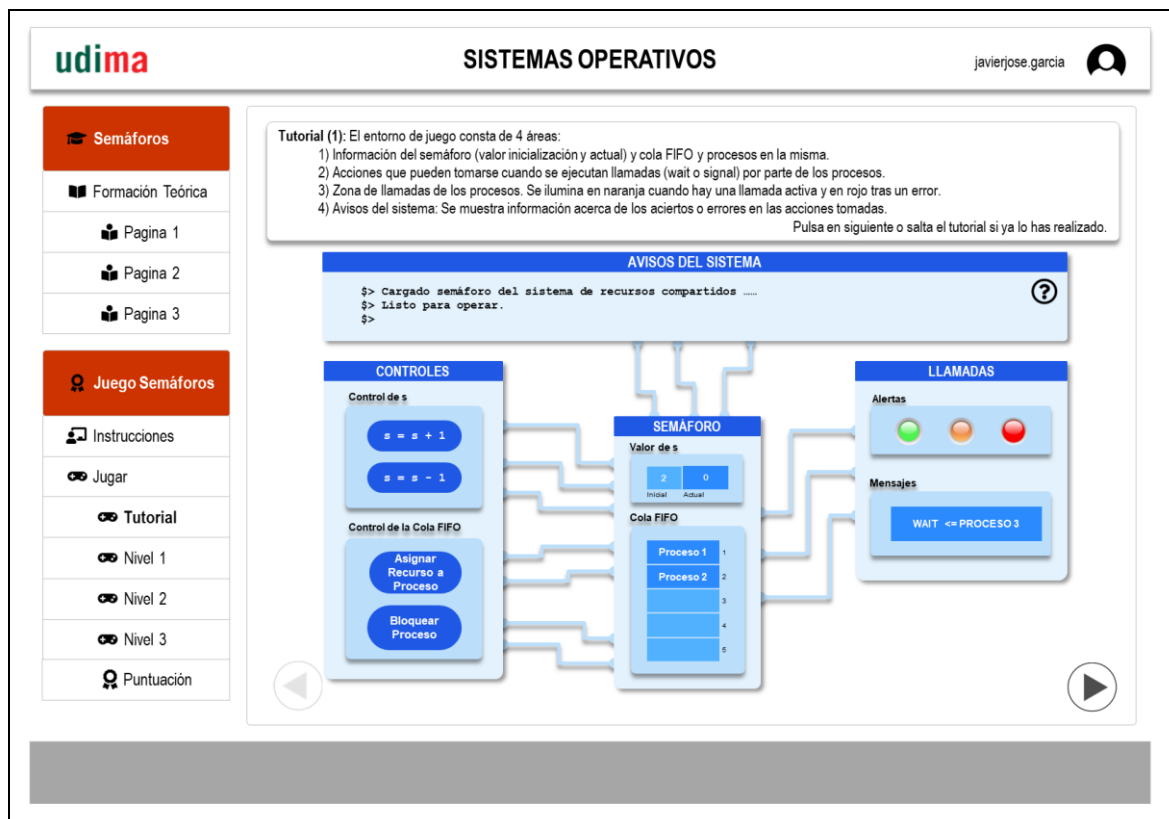


Figura 13: Rediseño de la Aplicación de Semáforos.

4.8. Implementación.

Una vez completado el rediseño y antes de entrar propiamente en la etapa de implementación, quedan por detallar algunas tareas previas.

4.8.1. Estructura de la Plataforma.

Para poder implementar la plataforma adecuadamente, mediante el diseño MVC (Model View Controller) y siguiendo las pautas de la formación de CodeIgniter, se hacía necesario definir el controlador, los modelos de datos y la base de datos en la que se apoya todo el diseño. Aprovechando la capacidad de CodeIgniter de generar e interpretar rutas URI se definió el modelo de enrutamiento adecuado que permitiera gestionar la aplicación en curso, el área en el que se encontraba el usuario en cada momento y la página deseada y que tuviera una correspondencia directa con la base de datos que gestionaría todo.

El esquema de gestión se define en el fichero Routes.php y permite redirigir las llamadas *get* o *post* hacia diversas funciones del Controlador. Aun así era necesario establecer una redirección de carga directa para los ficheros auxiliares (ficheros de estilo, código JavaScript, iconos e imágenes, etc) que fuera necesario cargar en cada caso. El código necesario incorporado al fichero es el mostrado en la Figura 14.

```
$routes->get('apps/(:any)', 'Aplicaciones::control/$1');  
$routes->get('apps', 'Aplicaciones::index');  
$routes->match(['post'], 'apps/ajaxRequestPost', 'Aplicaciones::ajaxRequestPost');  
$routes->get('(:any)', 'Pages::showme/$1');
```

Figura 14: Fichero Routes.php

La estructura del sistema de aplicaciones quedaba oculta al usuario y se le mostraba una estructura URI coherente y sencilla de seguir e interpretar, de forma que la visión de la URL de cada página le da información de donde se encuentra exactamente.

4.8.2. Base de datos.

Para el diseño de la base de datos se partió de un estudio de las necesidades de información que había de proveer la misma a la plataforma. La estructura necesitaba de

diversas tablas y una vista para filtrar de forma sencilla y rápida las puntuaciones máximas para la tabla final de la aplicación.

En la Figura 15 se muestra el Modelo de Diseño Lógico de la misma y en la Figura 16 la estructura de tablas.

La escasez de tiempo sufrida durante el desarrollo de este trabajo no ha permitido completar la base de datos, como habría sido conveniente y deseable, ni realizar el estudio completo y correcto del modelo de diseño. De hecho, durante la implementación se detectaron carencias en el diseño que forzaron a una ampliación de algunos campos en las tablas de *menús* y *páginas*.

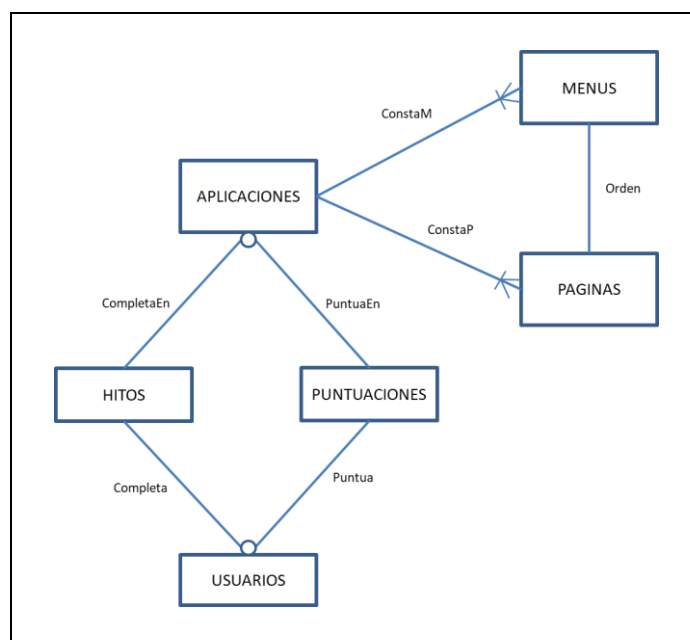


Figura 15: Modelo de Diseño Lógico de la Base de Datos.

El no haber podido completar el área de administración y/o gestión de la plataforma, como habría sido ideal, hace necesario introducir manualmente o mediante scripts los datos necesarios para operar la plataforma, pero no le resta ni efectividad ni eficiencia.

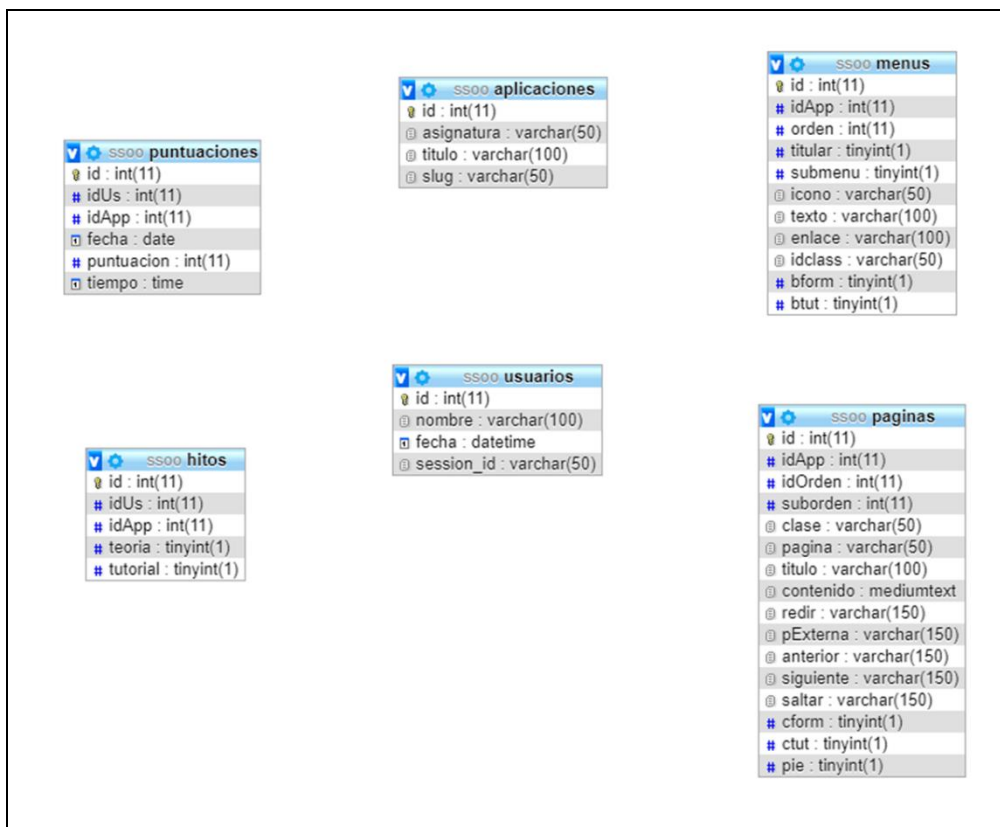


Figura 16: Estructura de Tablas de la Base de Datos

4.8.3. Control de Usuarios y de sesión.

Como ya se ha expresado en el apartado de Requisitos y Decisiones de desarrollo, el control de usuarios se diseñó de forma muy sencilla por lo que el control de la aplicación recae, únicamente, bajo el control de la sesión para que el usuario no pierda el control de avance y resultados en el juego,

CodeIgniter provee un control de sesión integrado en el framework que hace su trabajo perfectamente. Solo hay que configurar, de forma adecuada, los parámetros básicos y el tiempo de sesión que debe mantener el sistema. Durante el desarrollo y las pruebas se estableció el control de sesión con un tiempo de actualización de 2 horas, pero después

de los diversos tests y pruebas, un tiempo de 30 minutos parece adecuado y es el que se ha dejado establecido en los ajustes finales de CodeIgniter.

4.8.4. Controlador y Modelos de datos.

La última cuestión que hay que diseñar y definir antes de comenzar la implementación es el Controlador y los Modelos de Datos.

Por cuestiones de simplicidad, un controlador único es lo más adecuado dado que puede gestionar de forma sencilla varias aplicaciones. El controlador Aplicaciones.php consta de 5 funciones para desarrollar todo su cometido:

index: Se ocuparía de gestionar la aplicación de control o administración de existir. Dado que se ha quedado en el tintero, solo muestra una página de error para mantener la correcta apariencia de control de la aplicación.

control: Función principal del controlador. Gestiona la sesión y el usuario, comprueba que la página pedida exista en la base de datos y traspasa el trabajo de mostrar dicha página a la función **show**.

show: Encargada de cargar todos los datos necesarios para mostrar la página pedida de la base de datos y componer el resultado final que se muestra en el navegador.

gestUser: Función que se ocupa de la gestión de los usuarios de la aplicación.

ajaxRequestPost: Función que recibe los datos de puntuación enviados por jQuery y los traslada al modelo encargado de gestionar ese apartado.

Para realizar su trabajo, el controlador se apoya en 7 modelos de datos que regulan el flujo de entrada y salida de la base de datos. El detalle de los mismos se muestra en la

Tabla 6 y en el Anexo 4 se pueden encontrar detalles completos del código de la plataforma.

Nombre	Flujo de Datos	Tabla de la BBDD
AppsModel	Lectura	aplicaciones
MenuModel	Lectura	menus
ContentModel	Lectura	paginas
SessionModel	Lectura/Escritura	usuarios
ProgressModel	Lectura/Escritura	hitos
ScoreModel	Lectura/Escritura	puntuaciones
MaxscoreModel	Lectura	maxscores

Tabla 6: Modelos de datos

La ventaja añadida es que el modelo de datos de Codeigniter se ocupa de la gestión de la conexión a la base de datos, la securización de la misma y la gestión de lectura y escritura. La cantidad de líneas de código a escribir para estas tareas es mínima, lo que evita errores y simplifica el trabajo.

4.8.5. Implementación de la plataforma.

Con todo lo anterior iniciado se procedió a escribir el código del nucleo de la plataforma. Para simplificar el trabajo y eliminar problemas se dividió el total de la página en 5 bloques, de manera que cada uno cumple una función sencilla:

Cabecera: Corresponde al head de la página. Incluye las hojas de estilo, el icono de la página y el título obtenido de la base de datos.

Título: Cabecera de la página (área superior de icono, asignatura y usuario).

Menu: Gestión completa del área de menú, obtenido de la base de datos. Todo el contenido se genera dinámicamente usando hojas de estilo.

Cuerpo: Parte principal de la web. El contenido se carga de la base de datos o de páginas externas, según el caso. Incluye el control de los botones de adelante y atrás.

Pie: Pie de la página con el copyright y el cierre de todas las etiquetas previas.

Por supuesto, para la gestión de los estilos se han utilizado dos hojas de estilo en css, una para el todo el estilo general de las aplicaciones (*apps.css*) y otra para el contenido y estructura de los juegos (*game.css*).

4.8.6. Las páginas estáticas de las aplicaciones.

Las páginas estáticas de las aplicaciones, teoría, instrucciones y tutorial, se encuentran almacenadas en la base de datos dado que son muy breves por la propia implementación de la plataforma.

Se ha mantenido un estilo sencillo basado en cuadros con ligera sombra que destacan lo importante sin añadir complejidad ni sobrecargar la página.

Los botones de adelante y atrás se gestionan como apuntes en la base de datos para que sea totalmente automatizado su control. En la propia base de datos se proporcionan, en formato URI, los enlaces a los destinos de dichos botones, lo que hace que la aplicación sea sencilla de mantener y gestionar.

4.8.7. Los Juegos y el JavaScript.

Los juegos se han programado en JavaScript, cada uno en un fichero js, lo que permite modificarlos o ampliarlos de forma sencilla y sin tocar la estructura principal.

Constan de varias funciones que realizan cada una de las tareas (gestión de colores, gestión de leds, control de botones, etc). Es el usuario el que inicia el procesamiento del JavaScript mediante el botón “Iniciar nivel” presente en cada uno de los juegos. Se ha procurado realizar el código de forma clara y estructurada, aunque eso haya supuesto incrementar ligeramente la extensión del código.

Todo el control de los juegos se realiza en base a procesos aleatorios para generar las llamadas de los semáforos o crear los datos de los problemas. En el caso de los semáforos se limitó, por cuestión de estética, el tamaño de la cola a 5 elementos y, por tanto, el generador aleatorio está “controlado” para evitar que genere colas FIFO mayores.

Para los problemas de rendimiento se marcaron limitaciones y rangos válidos para no permitir problemas excesivamente complejos o sin sentido. Las respuestas erróneas se calculan con base en las mismas fórmulas correctas pero usando errores habituales en la resolución de este tipo de problemas. Aun así, y ante la posibilidad de que puedan dar alguna respuesta muy cercana a la realidad, se controla ese extremo para evitar que dos de las opciones tengan la misma respuesta.

Al finalizar la partida, la librería jQuery es la encargada de enviar los datos a la base de datos de la aplicación, a la función *ajaxRequestQuery*, y redirigir la web a la pantalla de resultados, donde el usuario tendrá a su disposición la puntuación obtenida y el cuadro de mejores puntuaciones del sistema.

Al finalizar la implementación se estudió la posibilidad de introducir en el juego de semáforos un botón de ayuda que mostrara el código del semáforo durante la misma, haciendo más sencillo el avance en la misma sin necesidad de contar con el código en papel o tener que abrirlo en una nueva ventana. Puesto que no era una implementación complicada y se había pedido por uno de los participantes del Test de Usabilidad, se decidió hacerlo.

4.8.8. El resultado final.

Una vez implementado todo de acuerdo a lo explicado en los apartados anteriores, el resultado final puede verse en las Figuras 17 y 18.

Los iconos de las distintas partes del menú se sustituyeron a última hora para intentar hacer más entendible y variado el menú, pero se han seguido usando los iconos de FontAwesome.

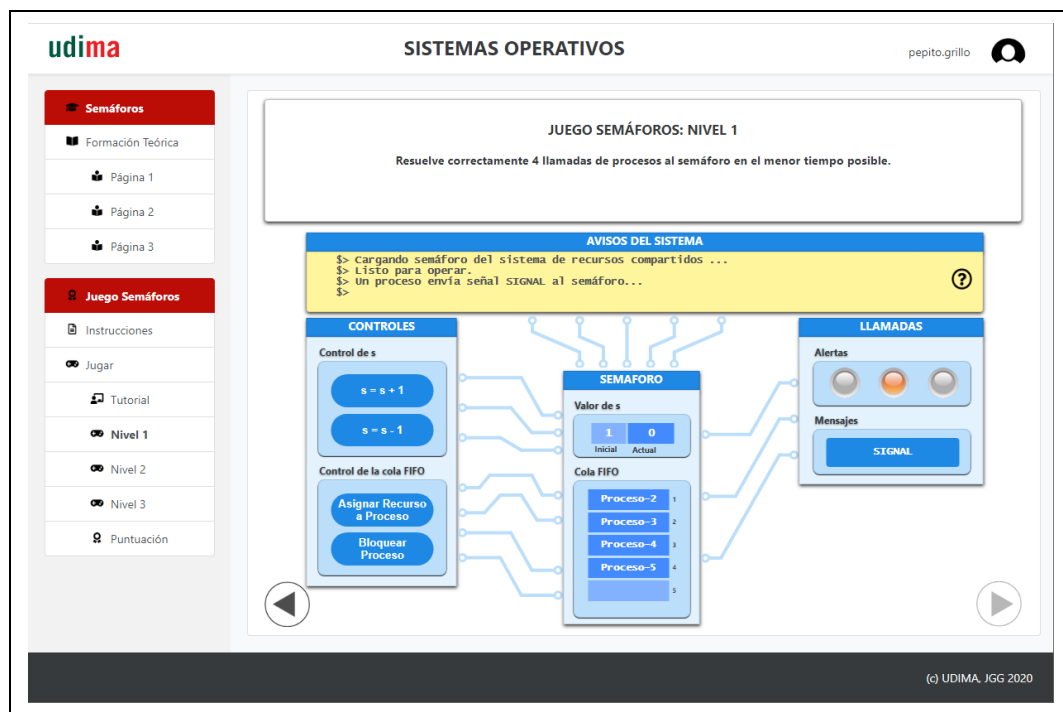


Figura 17: Aplicación de Semáforos final.

4.8.9. Compatibilidad.

En el desarrollo de páginas y aplicaciones web es una constante el descubrir que los distintos navegadores no se comportan de la misma forma en la interpretación del código y es uno de los problemas habituales que trae de cabeza a los desarrolladores.

Dentro de la limitación de posibilidades de prueba de las que se disponía, se ha intentado asegurar la máxima compatibilidad con los navegadores habituales.

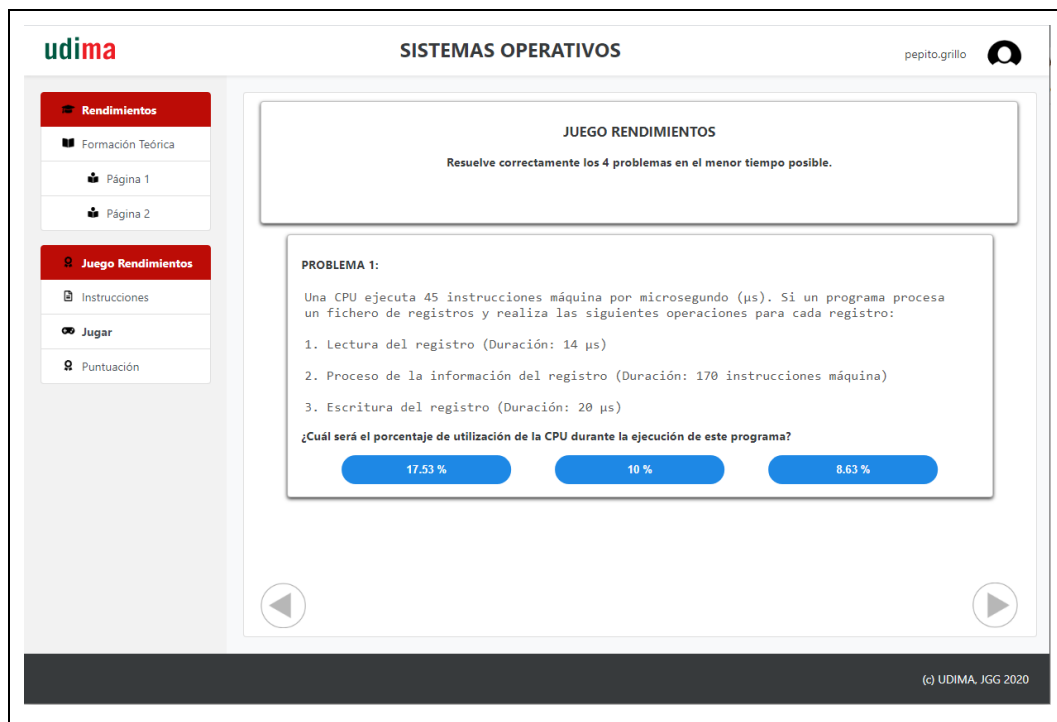


Figura 18: Aplicación de Rendimientos final.

Durante el desarrollo se realizaban pruebas constantemente usando Chrome, Firefox y Edge y, en cuanto estuvieron medianamente terminadas las aplicaciones, se realizaron una serie de test en Safari.

Pese a que se ha trabajado intentando mantener los estándares marcados por la W3C, en ocasiones ha sido necesario utilizar etiquetas CSS específicas para algún navegador en concreto o eliminar pequeños ajustes porque su comportamiento no era adecuado en todos los navegadores y no daba impresión de solidez en el diseño.

En cualquier caso, se necesita mucho tiempo de ajustes y pruebas para lograr mantener una correcta compatibilidad con todos los navegadores habituales, lo que encarece la implementación de la aplicación.

CAPÍTULO 5: EVALUACIÓN

Una vez terminada la implementación, aunque en realidad estaba a falta de algunos ajustes en los CSS y correcciones menores, había que pasar a evaluar la plataforma y la implementación.

Por un lado se optó por realizar una primera evaluación por expertos, en éste caso la profesora de la asignatura, con idea de validar tanto el contenido teórico como la operativa de las aplicaciones lúdicas.

Por el otro, se llevó a cabo el segundo Test de Usabilidad con una parte de los participantes en el primero, aunque esta vez se realizó de forma síncrona para poder calibrar de forma directa la reacción de los usuarios a la implementación real.

5.1. Evaluación por expertos.

Mediante sesión síncrona, se llevó a cabo el Test de Evaluación por Expertos. Las instrucciones para el test realizado eran las mismas que las que posteriormente se usarían en el Segundo Test de Usabilidad ya que la operativa era exactamente la misma, aunque con intención de detenerse más a fondo en los contenidos teóricos de las plataformas y validar que podrían cumplir su objetivo de servir de apoyo formativo a los estudiantes de la asignatura de Sistemas Operativos.

La profesora de la asignatura analizó cuidadosamente la aplicación y destacó, entre otras cosas, lo limpio y claro del diseño y que el uso del enmarcado de los conceptos teóricos permitía centrar la atención en ellos.

Igualmente dio su aprobación a los contenidos teóricos y alabó la sencillez y facilidad de entendimiento de la alegoría del taller mecánico en la Aplicación de Semáforos, que

hacía muy entendible la operativa mediante el uso de un ejemplo de fácil comprensión para el alumno.

En la Aplicación de Rendimientos destacó la presentación de los resúmenes estilo algoritmo que permitían aplicar los conocimientos a la resolución de problemas de los tipos presentados.

Respecto a los juegos, hizo incapié en la presentación y sencillez de operativa y disfrutó de unas partidas en las que puso a prueba sus conocimientos de los semáforos. Durante la partida puede observar que la modificación llevada a cabo en el código del semáforo para hacerlo más “entendible” e introducir una variación curiosa al juego, no resultaba tal y como yo había planeado y decidí estudiar el comportamiento de los participantes en el Segundo Test de Usabilidad para valorar si merecía la pena volver al código clásico del semáforo.

5.2. Segundo Test de Usabilidad.

La imposibilidad de encontrar tiempo material para realizar 13 tests síncronos obligó a reducir el número de participantes a un volumen más manejable pero que garantizara la viabilidad y fiabilidad del test. Así pues la cifra se redujo a 7.

Se envió aviso a todos los integrantes de la primera prueba y se decidió que el azar de los primeros 7 que respondieran sería el que hiciera la elección. Un pequeño extracto de las instrucciones enviadas a los participantes se puede ver en la Figura 19, aunque el detalle completo de las instrucciones y el cuestionario post-test puede ser consultado en el Anexo 5.

Instrucciones del Segundo Test de Usabilidad de las Aplicaciones de Semáforos y Rendimientos para la asignatura de Sistemas Operativos.

Gracias por tu ayuda en la evaluación de los prototipos de las aplicaciones de apoyo a la formación de la asignatura de Sistemas Operativos.

En el Test se desea comprobar la usabilidad de la implementación en la plataforma final de las aplicaciones y, sobre todo, la jugabilidad de las aplicaciones formativas. Dado que se han implementado mejoras y cambios de acuerdo a los resultados del primer test, deberemos repasar los apartados ya revisados en el primer test pero nos centraremos en el apartado de los juegos que no se incluía en el primer test.

Ten en cuenta que este test no se realiza sobre una maqueta gráfica sino sobre la plataforma final, aunque aún se encuentre en desarrollo. Para simplificar la operativa, se han activado previamente todas las funciones que se activan al completar ciertas acciones.

Una vez hayas completado las tareas te haré una pequeña batería de preguntas para valorar tus sensaciones sobre el entorno y el test. Si prefieres contestarlas con calma puedo enviarte la encuesta y me la envías una vez completada.

Para la realización del test debes suponer que estas accediendo a la formación online, a través de la plataforma Moodle de la Udim, de los temas de semáforos y rendimientos. Las tareas que has de realizar son:

- 1- Usa el enlace a continuación para acceder a la formación online:

<https://ssoo.anraingenieros.es/juegosSSOO.php>

Figura 19: Detalle Instrucciones Segundo Test de Usabilidad.

Durante los tests los participantes compartieron su escritorio para que pudiera observar su desempeño mientras iban realizando los pasos indicados. Todos los participantes pudieron realizar el test con fluidez y sin ningún problema digno de mención.

Todos los participantes, sin excepción, alabaron la claridad y limpieza de la implementación y quedaron encantados con el rediseño de los juegos.

Sin embargo, y como no podía faltar en éste tipo de tests, se localizaron, por ambas partes, varios “errores” o pequeños problemas que han requerido una intervención posterior en la plataforma para su corrección antes de dar la misma por cerrada:

- Se observó que casi todos los participantes dudaban o les resultaba extraño el comportamiento del semáforo ante la modificación de código introducido para hacer más “variado” el juego.
- En alguna versión del navegador Safari se produjeron pequeños desajustes en los controles del juego de semáforos.
- Se detectaron algunos pequeños desajustes en los CSS si el usuario tenía modificadas las fuentes y tamaños de las mismas en el sistema.
- Se detectó una situación en la que el juego de semáforos aceptaba una entrada “parcialmente correcta” debido a la modificación del código.

Por último, en el cuestionario posterior al Test, todos los usuarios de la plataforma coincidieron en que los diseños habían mejorado sustancialmente desde el primer test, haciendo la plataforma más entendible y mejor integrada. Asimismo alabaron la jugabilidad y coincidieron en que las aplicaciones podrían resultar muy útiles para su uso en la asignatura de Sistemas Operativos.

5.3. Ajustes finales.

Después de analizar los resultados del segundo test, era obvio que no podía cerrarse la aplicación sin acometer las modificaciones necesarias a los problemas localizados en el mismo.

Así, se optó por sustituir el código del semáforo por el mostrado en el manual de la asignatura (aunque realmente ordenado de forma inversa) y cambiar el comportamiento del juego. Además, se revisó todo el contenido de las hojas de estilo para mejorar la compatibilidad, en la medida de lo posible, con ajustes personalizados de fuentes y tamaños y evitar descuadres y se realizaron nuevas pruebas en Safari para garantizar una compatibilidad adecuada.

CAPÍTULO 6: CONCLUSIONES

6.1. Conclusiones en relación a los objetivos.

El haber centrado el ciclo de vida del desarrollo de las aplicaciones en un prototipado evolutivo ha permitido completar los objetivos propuestos logrando una implementación con un alto grado de usabilidad y que cumple con los atributos ya reseñados:

Facilidad de aprendizaje: El contenido teórico y lúdico propuesto, tal y como han señalado todos los participantes en las distintas evaluaciones y tests, permite a los que lo utilizan aprender de una forma sencilla conceptos formativos importantes para un estudiante de Sistemas Operativos y de un Grado de Ingeniería Informática en general.

Recuerdo en el tiempo: La metodología de ludificación o gamificación hace que los conceptos aprendidos mediante la plataforma se fijen de forma duradera ya que el recuerdo del concepto se asocia al recuerdo de la operativa del juego.

Eficiencia de uso: Todos los participantes en las pruebas llevadas a cabo para el desarrollo del sistema han destacado la sencillez, limpieza y facilidad de uso y comprensión del sistema. El haber basado el esquema operativo en un entorno conocido ha reducido enormemente la dificultad de entendimiento del manejo y ha permitido que sean innecesarias instrucciones complejas o explicaciones largas.

Baja tasa de errores: La eficiencia de uso se ha traducido en una baja tasa de errores. Casi nadie ha tenido errores de manejo de la plataforma y, en los juegos, los errores han sido más debidos a falta de soltura con el recuerdo del código de los semáforos que a problemas de la plataforma. De hecho, aquellos que se animaron a realizar una segunda

partida en la plataforma descubrieron que ya no necesitaban consultar la ayuda y la respuesta salía casi sola de sus cabezas.

Satisfacción obtenida del sistema: A la vista de los comentarios recibidos durante las pruebas, la plataforma parece proporcionar una satisfacción muy alta a los que la utilizan y, de hecho, la respuesta a la pregunta realizada en el segundo test de usabilidad: “¿Crees que si no supieras nada de Semaforos y rendimientos habrías aprendido algo con las plataformas?” obtuvo una respuesta unánime de “Mucho”.

En definitiva, el modelo de desarrollo seleccionado ha resultado fundamental para alcanzar, con suficiente calidad, los objetivos propuestos. La información recibida, sobre todo en el primer test, de los participantes, ha resultado fundamental para mejorar el diseño y la usabilidad de los prototipos iniciales. Esto ha permitido obtener una implementación final que ha superado con creces mis expectativas más optimistas.

Respecto a los objetivos secundarios fijados en el apartado 1.2 de este trabajo, parece adecuado aseverar que se ha logrado su cumplimiento.

Para alguien que solo había realizado pequeños trabajos en PHP, HTML o CSS, desarrollar toda una plataforma con un estilo eficaz, limpio, agradable e integrado, ha supuesto un reto. Sin embargo, también ha sido enormemente formativo, sobre todo si se une al descubrimiento de la propia capacidad para llevar a cabo interfaces agradables y usables.

El desarrollo como una plataforma de aplicaciones que se fijó como objetivo secundario, pese a ser muy ambicioso, ha permitido enfocar el conjunto desde una perspectiva diferente y ha facilitado muchísimo la implementación de las aplicaciones una vez que la plataforma estaba encajada. De hecho, llevar a cabo las modificaciones finales indicadas en el Capítulo 5 ha supuesto solo unos pocos minutos de trabajo.

Por último, como ya se mencionó en el apartado 3.1 y se muestra a continuación en la Figura 20, el análisis a posteriori de la utilización de los recursos temporales a lo largo del desarrollo frente a la Planificación Temporal inicial ha mostrado una mínima desviación centrada, principalmente, en las áreas de Prototipado, demasiado pesimista, y Redacción, demasiado optimista, pero sin influencia grave en la planificación.

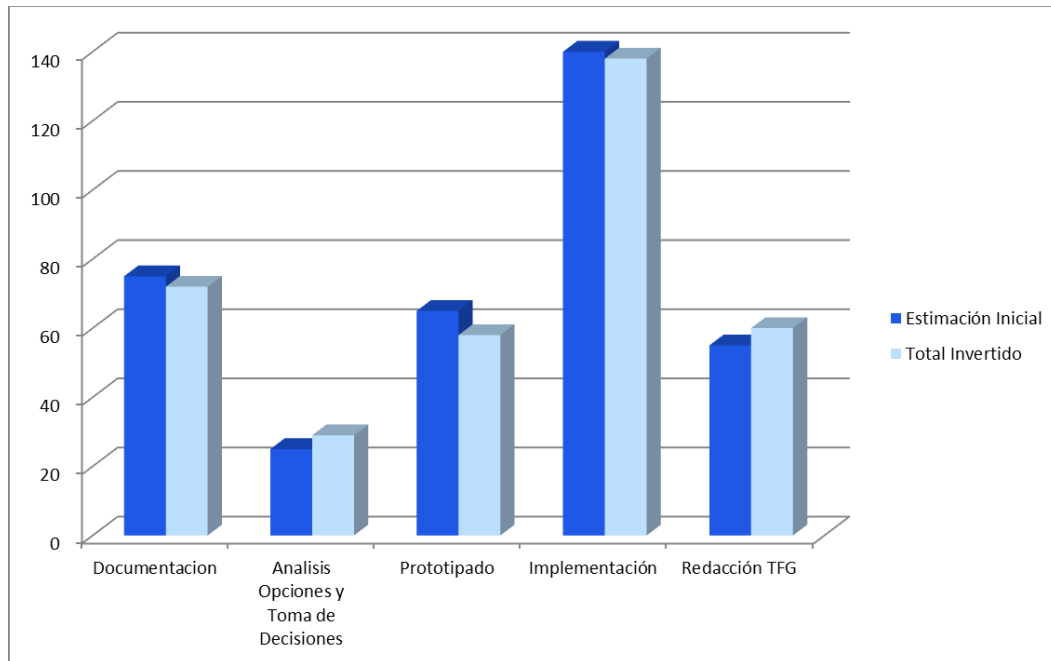


Figura 20: Comparación gráfica de la Planificación Temporal.

Cabe añadir, sin embargo, que la implementación final habría necesitado de algunas horas más de dedicación, sobre unas 8 o 10, para acabar de realizar ajustes de las hojas de estilo y detalles menores de la interfaz, así como reorganizar y comentar el contenido de los CSS.

6.2. Trabajos futuros propuestos.

A la vista de los resultados obtenidos, el planteamiento de una plataforma de aplicaciones formativas que sienta las bases a un sistema global de gestión de este tipo de aplicaciones puede resultar interesante, tanto desde el punto de vista de mantener un interfaz común y sencillo, como desde el de poder integrar la gestión del control de usuarios de la misma en un sistema global integrado con el Moodle universitario.

Respecto al área de Sistemas Operativos, y aunque no todo son juegos ni diversión, es más que probable que existan aun algunos epígrafes formativos que puedan mejorar su catálogo de soluciones con la incorporación de la ludificación o de soluciones con mayor carga lúdica que colaboren a mejorar el nivel formativo.

BIBLIOGRAFÍA

- [1] S. Pamplona, N. Medinilla, and P. Flores, “A Systematic Map for Improving Teaching and Learning in Undergraduate Operating Systems Courses,” *IEEE Access*, vol. 6, pp. 60974–60992, 2018, doi: 10.1109/ACCESS.2018.2871768.
- [2] M. M. A. Y. S. Baran, “Learning Physics through Project-Based Learning Game Techniques,” *Int. J. Instr.*, vol. 11, pp. 221–234, Apr. 2018, Accessed: May 07, 2020. [Online]. Available: <https://eric.ed.gov/?id=EJ1175029>.
- [3] J. M. D. Hill, C. K. Ray, J. R. S. Blair, and C. A. Carver, “Puzzles and games,” 2003, p. 182, doi: 10.1145/611892.611964.
- [4] S. Tang and M. Hanneghan, “Fusing games technology and pedagogy for games-based learning through a model driven approach,” in *2011 IEEE Colloquium on Humanities, Science and Engineering, CHUSER 2011*, 2011, pp. 380–385, doi: 10.1109/CHUSER.2011.6163756.
- [5] M. Peiro, “Los juegos de mesa mejoran el rendimiento y motivan al alumno | UDIMA,” 2019. <https://www.udima.es/es/primeras-jornadas-innovacion-universitaria-innovaudima-manifiesto-novedades-tecnologia-educativa.html> (accessed May 07, 2020).
- [6] J. D. Calvo, “Diseño e implementación de una aplicación en HTML 5 para el aprendizaje de la gestión de procesos en un sistema operativo.,” UDIMA, 2016.
- [7] M. J. Tesorero, “Diseño de un juego para el aprendizaje de la gestión de los procesos en un sistema operativo.,” UDIMA, 2016.
- [8] C. Manzanares, “Diseño e implementación de una aplicación software para el aprendizaje del funcionamiento del algoritmo de reemplazo LRU para sistemas con memoria virtual.,” UDIMA, 2019.
- [9] X. Ferre, *Interacción persona-ordenador*. Centro de Estudios Financieros (CEF), 2015.
- [10] F. D. de la Peña Esteban, J. A. Lara Torralbo, D. Lizcano Casas, and M. C. Burgos García, “Web gamification with problem simulators for teaching engineering,” *J. Comput. High. Educ.*, vol. 32, no. 1, pp. 135–161, Apr. 2020, doi: 10.1007/s12528-019-09221-2.
- [11] D. Dicheva, C. Dichev, G. Agre, and G. Angelova, “International Forum of Educational Technology & Society Gamification in Education: A Systematic Mapping Study Gamification in Education: A Systematic Mapping Study,” *Source J. Educ. Technol. Soc.*, vol. 18, no. 3, pp. 75–88, 2015, doi: 10.2307/jeductechsoci.18.3.75.
- [12] J. M. Garcia Platero, “CVC. Ludificación | Martes Neológico,” *Blog Centro Virtual Cervantes*, 2018. <https://blogscvc.cervantes.es/martes-neologico/ludificacion/> (accessed Jun. 03, 2020).

- [13] O. Borrás Gené, “Fundamentos de Gamificación,” Feb. 2017.
- [14] S. Pamplona, N. Medinilla, and P. Flores, “Exploring misconceptions of operating systems in an online course,” in *ACM International Conference Proceeding Series*, 2013, pp. 77–86, doi: 10.1145/2526968.2526977.
- [15] Y. B.-D. Kolikant, M. Ben-Ari, and S. Pollack, “The anthropology semaphores,” *ACM SIGCSE Bull.*, vol. 32, no. 3, pp. 21–24, Sep. 2000, doi: 10.1145/353519.343061.
- [16] “Semaphores are Surprisingly Versatile,” *Preshing on Programming Blog*, 2015. <https://preshing.com/20150316/semaphores-are-surprisingly-versatile/> (accessed Mar. 25, 2020).
- [17] S. Pamplona and J. Garrido, *Sistemas Operativos*. Centro de Estudios Financieros (CEF), 2016.
- [18] “REGLAMENTO (UE) 2016/ 679 DEL PARLAMENTO EUROPEO Y DEL CONSEJO - de 27 de abril de 2016 - relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/ 46/ CE (Reglamento general de protección de datos).”
- [19] Google, “Design - Material Design.” <https://material.io/design> (accessed May 06, 2020).
- [20] “CodeIgniter Web Framework.” <https://codeigniter.com/> (accessed Jun. 04, 2020).

ÍNDICE DE FIGURAS

Figura 1: Ciclo de vida del proyecto.	19
Figura 2: Código del semáforo.	22
Figura 3: Ejemplo didáctico de semáforo.....	22
Figura 4: Rendimiento en una planificación Round Robin	23
Figura 5: Porcentaje de utilización de la CPU.....	23
Figura 6: Página 1 de Formación Teórica de la Aplicación de Semáforos.....	26
Figura 7: Página 1 del Tutorial de la Aplicación de Semáforos	27
Figura 8: Detalle de la maqueta de la Aplicación de Semáforos en Marvel.....	28
Figura 9: Detalle de las Instrucciones del Primer Test de Usabilidad	29
Figura 10: Detalle del Formulario de Test a los participantes.	30
Figura 11: Valoración del Primer Test de Usabilidad: Completo.	32
Figura 12: Valoración del Primer Test de Usabilidad: Diseño.....	32
Figura 13: Rediseño de la Aplicación de Semáforos.....	38
Figura 14: Fichero Routes.php	39
Figura 15: Modelo de Diseño Lógico de la Base de Datos.	40
Figura 16: Estructura de Tablas de la Base de Datos.....	41
Figura 17: Aplicación de Semáforos final.	46
Figura 18: Aplicación de Rendimientos final.	47
Figura 19: Detalle Instrucciones Segundo Test de Usabilidad.	51
Figura 20: Comparación gráfica de la Planificación Temporal.	55

ÍNDICE DE TABLAS

Tabla 1: Planificación Inicial.	14
Tabla 2: Comparativa de las Planificaciones Temporales Inicial y Real.	14
Tabla 3: Presupuesto estimado.	16
Tabla 4: Comparativa de Costes de Alojamientos.	17
Tabla 5: Participantes en el Primer Test de Usabilidad	31
Tabla 6: Modelos de datos	43

ANEXOS

Anexo 1. Maqueta Inicial.

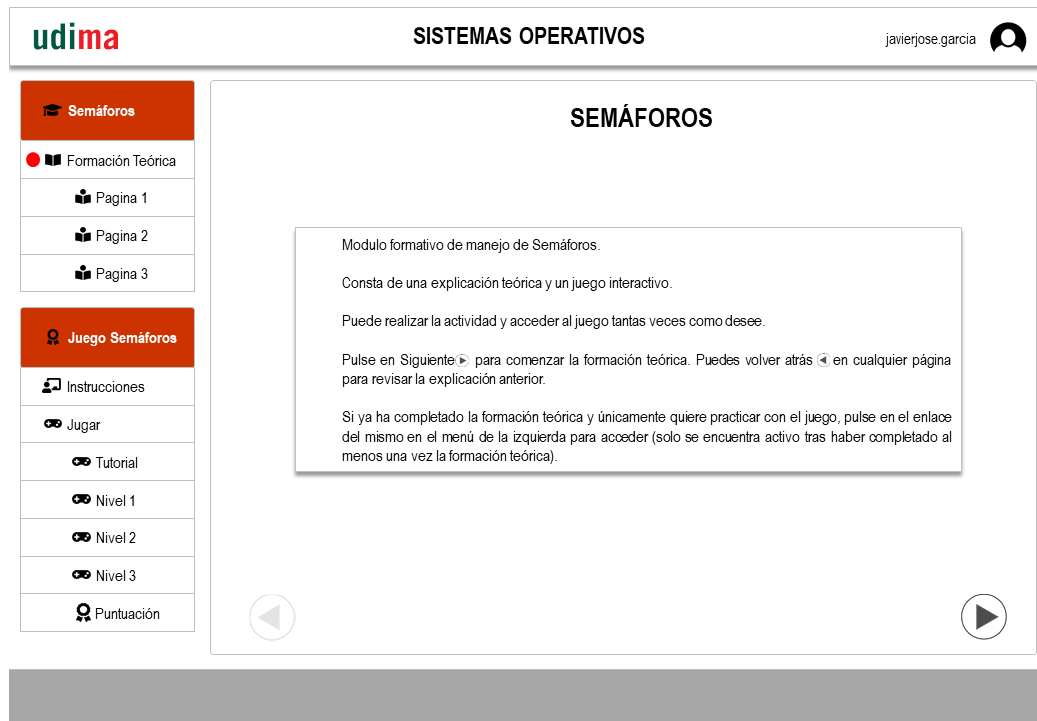
Anexo 2. Primer Test de Usabilidad.

Anexo 3. Respuestas al Primer Test de Usabilidad.

Anexo 4. Detalles del Código de la Plataforma.

Anexo 5. Segundo Test de Usabilidad.

ANEXO 1: MAQUETA INICIAL



udima

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

SEMÁFOROS

(Página 2)

El código de las operaciones es:

```

wait (s){
    if (s > 0)
        s = s - 1;
        Asignar recurso al proceso;
    else
        Bloquear al proceso;
}

signal (s){
    if (!procesos bloqueados)
        s = s + 1;
    else
        Desbloquear al siguiente proceso bloqueado en la cola por la
        operación wait;
}

```

Las operaciones wait y signal son indivisibles o atómicas. En cuanto a la operación signal esto quiere decir que el proceso que la ejecuta no puede ser interrumpido hasta que no ha finalizado la rutina. La operación wait es algo más compleja: si el semáforo es mayor que cero, se comprueba y decremента en una operación indivisible; sin embargo, si el semáforo tiene el valor cero, el proceso se bloquea.

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

SEMÁFOROS

(Página 3)

Para entenderlo de una forma visual:

Imagina un taller en el que trabajan varios mecánicos pero solo hay 2 destornilladores disponibles ($s = 2$). El encargado del material dispone de un bloc de notas para tomar nota de las peticiones de uso del destornillador (*cola FIFO*).

Cuando un mecánico necesita el destornillador, pide (*wait*) al encargado que le asigne uno. Si el encargado tiene alguno disponible en el almacén ($s > 0$) se lo entrega al mecánico y reduce en 1 el stock ($s = s - 1$). Si no tiene disponible ninguno ($s \leq 0$), apunta la petición en el bloc (*añade a la cola FIFO*) y deja al mecánico a la espera de que quede libre un destornillador (*proceso bloqueado*).

Cuando un mecánico termina de usar el destornillador, lo devuelve al almacén (*signal*). El encargado comprueba en el bloc si hay mecánicos a la espera de destornilladores (*cola FIFO*). Si hay alguno, le entrega el destornillador (*desbloqueo del proceso*) y lo elimina de la lista (*cola FIFO*). Si no hay nadie a la espera, aumenta el stock en 1 ($s = s + 1$).

Curso 2019-20 – Ingeniería Informática

63

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

SEMÁFOROS

(JUEGO)

INSTRUCCIONES:

Debes tomar el papel del gestor del semáforo y realizar las acciones adecuadas para proveer del recurso gestionado a los procesos que lo soliciten.

El nivel inicial es un tutorial que te irá guiando en el manejo del juego y explicando los pasos a dar para realizar tu misión con eficacia. Si ya lo has realizado, puedes saltarlo usando el botón ► situado en la esquina inferior derecha.

Los siguientes niveles pondrán a prueba tu destreza como gestor de semáforos y premiarán tu acierto y velocidad. Ten cuidado, los errores reducen tu puntuación final.

Pulsa en siguiente para acceder al Nivel Tutorial:

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

Tutorial (1): El entorno de juego consta de 4 áreas:

1) Información del semáforo (valor actual de s y, entre paréntesis, valor de inicialización) y cola FIFO y procesos en la misma.

2) Acciones que pueden tomarse cuando se ejecutan llamadas (wait o signal) por parte de los procesos.

3) Zona de llamadas de los procesos. Se ilumina en amarillo cuando hay una llamada activa.

4) Avisos del sistema: Se muestra información acerca de los aciertos o errores en las acciones tomadas.

Pulsa en siguiente o salta el tutorial si ya lo has realizado.

1 Semáforo

s (3) = 0

Cola FIFO

Proceso 1

Proceso 2

2 Acciones

s = s + 1

s = s - 1

Asignar Recurso a Proceso: 1

Bloquear Proceso

3 Llamadas

wait (Proceso 3)

signal

4 Avisos del Sistema

64

Trabajo de Fin de Grado

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

!

Semáforo

s (3) = 0

Cola FIFO

Proceso 1

Proceso 2

Acciones

s = s + 1

s = s - 1

Asignar Recurso a Proceso:

1

Bloquear Proceso

Avisos del Sistema

Llamadas

wait (Proceso 3)

signal

◀

▶

Tutorial (2): En la zona de semáforo hay que observar:

- Valor de inicialización del semáforo (en el ejemplo mostrado es de 3) es decir, número de recursos que gestiona.

- Valor actual del semáforo (en la caja), es decir, recursos disponibles en éste momento.

- Lista de procesos bloqueados en la cola. El proceso mas arriba es el primero en haber entrado a la misma.

Pulsa en Siguiente.

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

Semáforo

s (3) = 0

Cola FIFO

Proceso 1

Proceso 2

Acciones

s = s + 1

s = s - 1

Asignar Recurso a Proceso:

1

Bloquear Proceso

Avisos del Sistema

Llamadas

! wait (Proceso 3)

signal

◀

▶

Tutorial (3): Cuando un proceso necesita el recurso gestionado por el semáforo, envía un **wait** al mismo y el indicador de llamada se ilumina en amarillo.

El gestor del semáforo, es decir, tu, deberá ejecutar las acciones adecuadas para responder a la llamada recibida.

Pulsa en Siguiente.

Curso 2019-20 – Ingeniería Informática

65

 Puntuación

signal



● Puntuación

-> Recurso liberado asignado al Proceso 1. OK



SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

SEMÁFOROS

(JUEGO)

NIVEL 1:

Resuelve los retos que se presentan en el menor tiempo posible y sin cometer errores.

Tu destreza y rapidez marcaran tu puntuación.

Fíjate bien en los valores de inicio y actuales del semáforo antes de comenzar el nivel y se cuidadoso.

Pulsa en Iniciar para comenzar el nivel:

INICIAR NIVEL

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

NIVEL 1

Semáforo

s (3) = 0

Cola FIFO

Proceso 2

Acciones

s = s + 1

s = s - 1

Asignar Recurso a Proceso: 1

Bloquear Proceso

Llamadas

wait

signal

Avisos del Sistema

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Semáforos

Formación Teórica

Página 1

Página 2

Página 3

Juego Semáforos

Instrucciones

Jugar

Tutorial

Nivel 1

Nivel 2

Nivel 3

Puntuación

¡¡¡JUEGO COMPLETADO!!!

VOLVER A JUGAR

Mejores Resultados

Nombre	Errores	Tiempo
fulanito.pi	0	0:11:23
pepe.perez	1	0:10:54
juancarlos.marzo	1	0:15:22
miguel.miguel	3	0:09:36
anton.anton	4	0:15:15

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Rendimientos

Formación Teórica

Página 1

Página 2

Juego Rendimientos

Instrucciones

Jugar

Nivel 1

Nivel 2

Nivel 3

Puntuación

RENDIMIENTOS

Modulo formativo de cálculo de Rendimientos.

Consta de una explicación teórica y un juego interactivo.

Puede realizar la actividad y acceder al juego tantas veces como desee.


Pulse en Siguiente para comenzar la formación teórica.

Si ya ha completado la formación teórica y únicamente quiere practicar con el juego, pulse en el enlace del mismo en el menú de la izquierda para acceder (solo se encuentra activo tras haber completado al menos una vez la formación teórica).

70


Trabajo de Fin de Grado

Rendimientos

 Formación Teórica

 Pagina 1 Pagina 2

 Juego Rendimientos

 Instrucciones

 Jugar

 Nivel 1

 Nivel 2

 Nivel 3

 Puntuación

RENDIMIENTOS

(Pagina 1)


En una planificación Round Robin el **Rendimiento** se calcula como:

Rendimiento = número de procesos / tiempo que tardan en ejecutarse los procesos

Método de cálculo:

1. Calcular las **rodajas** de cada proceso en función del tiempo por rodaja.
2. Calcular el número de **cambios de contexto** como rodajas - 1.
3. Sumar los tiempos de los dos pasos anteriores para obtener el tiempo total de proceso.
4. Calcular el **rendimiento** en μs según la fórmula anterior.

Rendimientos

 Formación Teórica

 Pagina 1 Pagina 2

Juego Rendimientos

 Instrucciones

 Jugar

 Nivel 1

 Nivel 2

 Nivel 3

 Puntuación

RENDIMIENTOS

(Pagina 2)

El **Porcentaje de Utilización de la CPU** se calcula como:

Porcentaje de Utilización = Tiempo que la CPU permanece ocupada / Tiempo total de ejecución

Método de cálculo:

1. Calcular el tiempo de cada operación.
2. Calcular el tiempo total del programa.
3. Identificar que operaciones ocupan a la CPU.
4. Calcular el **porcentaje de utilización de la CPU** (%) según la fórmula anterior.

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Rendimientos

Formación Teórica

Página 1

Página 2

Juego Rendimientos

Instrucciones

Jugar

Puntuación

RENDIMIENTOS

(JUEGO)

INSTRUCCIONES:

La empresa para la que trabajas comercializa un sistema operativo de última generación. Tu papel en la empresa es calcular los comportamientos de dicho sistema operativo en diversas situaciones para que los compañeros de programación puedan optimizar su funcionamiento.

Por cada problema correctamente resuelto, tanto de rendimiento como de utilización de CPU, y que ayude a mejorar el comportamiento del sistema operativo, la empresa te pagará un bonus de 500€. Sin embargo, los fallos minoran el bonus a obtener en ese problema.

Si consigues reunir 1.500€ de bonus con 4 problemas resueltos o menos, ascenderás a jefe de departamento.

Resuelve los problemas que te plantean y logra ascender en el menor tiempo posible.

INICIAR JUEGO

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Rendimientos

Formación Teórica

Página 1

Página 2

Juego Rendimientos

Instrucciones

Jugar

Puntuación

PROBLEMA 1:

Supongamos un sistema operativo con una planificación round-robin con un tiempo de $0,25 \mu s$ entre dos rodajas de tiempo (independientemente de si ocurre o no un cambio de proceso). Si se ejecutan tres trabajos de $6 \mu s$, $3 \mu s$ y $8 \mu s$. Contesta las siguientes preguntas:

a) ¿Cuál es el rendimiento si la rodaja de tiempo es de $1 \mu s$?

b) ¿Cuál es el rendimiento si la rodaja de tiempo es de $0,5 \mu s$?

c) ¿En cuál de los dos casos presentados anteriormente el rendimiento es mayor?

Nota: El rendimiento es el número de procesos ejecutados por unidad de tiempo.

Nota: Los resultados deben introducirse con coma decimal y dos decimales.

Pregunta	Respuesta
a)	<input type="text"/> μs
b)	<input type="text"/> μs
c)	<input type="text"/>

Enviar Respuestas

Bonus obtenido:

0 €

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Rendimientos

Formación Teórica

Página 1

Página 2

Juego Rendimientos

Instrucciones

Jugar

Puntuación

PROBLEMA 1:

Supongamos un sistema operativo con una planificación round-robin con un tiempo de 0,25 μ s entre dos rodajas de tiempo (independientemente de si ocurre o no un cambio de proceso). Si se ejecutan tres trabajos de 6 μ s, 3 μ s y 8 μ s. Contesta las siguiente preguntas:

a) ¿Cuál es el rendimiento si la rodaja de tiempo es de 1 μ s?

b) ¿Cuál es el rendimiento si la rodaja de tiempo es de 0,5 μ s?

c) ¿En cuál de los dos casos presentados anteriormente el rendimiento es mayor?

Nota: El rendimiento es el número de procesos ejecutados por unidad de tiempo.
Nota: Los resultados deben introducirse con coma decimal y dos decimales.

Pregunta	Respuesta
a)	0,14 μ s
b)	0,12 μ s
c)	a

Enviar Respuestas

Bonus obtenido: 0 €

udima

SISTEMAS OPERATIVOS

javierjose.garcia

Rendimientos

Formación Teórica

Página 1

Página 2

Juego Rendimientos

Instrucciones

Jugar

Puntuación

RESULTADOS PROBLEMA 1:

Pregunta	Respuesta	
a)	0,14 μ s	✓
b)	0,12 μ s	✓
c)	a	✓

!!! ENHORABUENA!!!

Has tenido: 3 aciertos y 0 fallos.

El bonus obtenido ha sido de: 500 €.

Siguiente Problema

Bonus obtenido: 500 €



Rendimientos

Formación Teórica

Página 1

Página 2

Juego Rendimientos

Instrucciones

Jugar

Puntuación

!!! ENHORABUENA!!!

Has logrado reunir un bonus de: 2.000 €

Te acaban de ascender a Jefe de Departamento.

Mejores Resultados

Puesto	Nombre	Bonus	Tiempo
1	fulanito.pi	2.000 €	0:05:45
2	mengano.perez	1.700 €	0:04:22
3	juancarlos.marzo	1.700 €	0:05:01
4	miguel.miguel	1.500 €	0:10:12
5	jorge.jordan	1.500 €	0:12:55

Volver a Jugar

ANEXO 2: PRIMER TEST DE USABILIDAD

Instrucciones del Primer Test de Usabilidad de las Aplicaciones de Semáforos y Rendimientos para la asignatura de Sistemas Operativos.

Gracias por tu ayuda en la evaluación de los prototipos de las aplicaciones de apoyo a la formación de la asignatura de Sistemas Operativos.

En éste test se desea comprobar la usabilidad del diseño de las aplicaciones y el contenido de los apartados de Teoría e Instrucciones o Tutorial de los juegos. En ésta versión no se puede jugar aún y la pantalla del juego es únicamente para mostrar la apariencia, pulsando en cualquier punto de ella se pasa a la pantalla final del juego.

El objetivo del test es identificar problemas en la estructura del diseño de las aplicaciones y medir su grado de usabilidad por parte de los usuarios potenciales. Ten en cuenta que se trata de una maqueta gráfica y que no te encuentras en un entorno web real, por lo que las funciones que se “activan” o “desactivan” en función de si ha sido realizado antes o no (como rezan los textos explicativos) se encuentran siempre activadas dando por supuesto que ya han sido realizadas antes.

Para la realización del test debes suponer que estas accediendo a la formación online, a través de la plataforma Moodle de la Udim, de los temas de semáforos y rendimientos. Las tareas que has de realizar son:

- 1- Usa el enlace a continuación para acceder a la formación online de Semáforos:

<https://marvelapp.com/e5cfai7>

- 2- Lee la información de la página de introducción y completa el recorrido por el módulo de formación teórica usando los botones adelante y atrás para pasar las páginas del módulo.
- 3- Usando los mismos controles, accede a la página de instrucciones del juego y completa el tutorial del mismo.
- 4- Accede ahora, al primer nivel del juego. Desgraciadamente, aun no es posible jugar al mismo, así que pulsando en cualquier parte de la ventana del juego accederás a la pantalla de puntuación final.
- 5- Imagina que el juego te ha gustado y quieres volver a jugar e intentar mejorar tu puntuación.
- 6- Completa el cuestionario, cuyo enlace se proporciona a continuación, y apórtame tus comentarios y valoraciones, de la forma más detallada y explícita posible, de tu experiencia con la plataforma:

<https://forms.gle/7Sn8Gi2GswjSjmDq6>

- 7- Ahora has de acceder a la formación online de Rendimientos mediante el enlace:

<https://marvelapp.com/e5d82i4>

- 8- Al igual que en el módulo anterior, lee la información de la página de introducción y completa el recorrido por el módulo de formación teórica con los controles de desplazamiento de página.
- 9- Siguiendo con el mismo formato y recorrido, accede a la página de instrucciones del juego y, tras leerlas, comienza con una partida.
- 10- Como la maqueta no es interactiva, aun, pulsando en cualquier parte de la pantalla del primer problema se te proporcionan las respuestas en las casillas adecuadas para que observes el comportamiento del juego.
- 11- Ahora puedes enviar las respuestas para que el sistema las compruebe. Observa tus resultados.
- 12- Como el juego no está aún operativo, aunque pulses en el botón “Siguiente Problema” la maqueta te lleva a la pantalla final de resultados para que puedas ver tus resultados y las puntuaciones.
- 13- Completa ahora, como en el módulo anterior, el cuestionario con tu experiencia con el uso de la plataforma y no olvide ser lo más detallado posible:

<https://forms.gle/w4zMCLmxgNe6ckpq5>

Espero que la experiencia haya sido satisfactoria y te agradezco enormemente tu colaboración.

Muchas gracias.

Javier José García García

Primer Test de Usabilidad de la Aplicación de Semáforos para Sistemas Operativos

*Obligatorio

1. Dirección de correo electrónico *

Tareas del test

2. ¿Has podido completar las tareas del test de usabilidad sin problemas? *

Marca solo un óvalo.

- ☐ Si
☐ Si, pero con problemas o dudas
☐ No

3. Si has tenido problemas en la realización del test, indica a continuación cuales.

4. Valora el nivel de dificultad del test y sus instrucciones. *

Marca solo un óvalo.

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

Diseño

Valora el diseño de la aplicación.

5. Valoración Personal *

Marca solo un óvalo.

	1	2	3	4	5	
No me gusta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Me encanta

6. Claridad y Limpieza *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

7. Funcionalidad *

Marca solo un óvalo.

	1	2	3	4	5	
Nada funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy funcional

8. ¿Qué le sobra o qué le falta? *

9. Comentarios

Contenido teórico

Valora el contenido del módulo de formación teórico de la aplicación.

10. Exactitud *

Marca solo un óvalo.

	1	2	3	4	5	
Incorrecto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Correcto

11. Claridad *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

12. Concisión *

Marca solo un óvalo.

	1	2	3	4	5	
Muy escueto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy extenso

13. ¿Añadirías algo? *

14. ¿Quitarías algo? *

15. Comentarios

Tutorial

Valora el tutorial de formación de uso del juego.

16. Claridad *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

17. Utilidad *

Marca solo un óvalo.

	1	2	3	4	5	
No es nada útil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Es muy útil

18. Extensión *

Marca solo un óvalo.

	1	2	3	4	5	
Es demasiado breve	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Es demasiado largo

19. ¿Qué le sobra o qué le falta? *

20. Comentarios

Diseño del
Juego

Valora el diseño del juego y su usabilidad (a partir de lo que has visto y lo que explica el tutorial)

21. Valoración personal *

Marca solo un óvalo.

	1	2	3	4	5	
No me gusta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Me gusta mucho

22. Claridad *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

23. Funcionalidad *

Marca solo un óvalo.

	1	2	3	4	5	
No parece funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Parece muy funcional

24. ¿Cambiarías algo? ¿Qué? *

25. Comentarios

Gracias por tu colaboración.

Este contenido no ha sido creado ni aprobado por Google.

Google Formularios

Primer Test de Usabilidad de la Aplicación de Rendimientos para Sistemas Operativos

**Obligatorio*

1. Dirección de correo electrónico *

Tareas del test

2. ¿Has podido completar las tareas del test de usabilidad sin problemas? *

Marca solo un óvalo.

- ☐ Si
- ☐ Si, pero con problemas o dudas
- ☐ No

3. Si has tenido problemas en la realización del test, indica a continuación cuales.

4. Valora el nivel de dificultad del test y sus instrucciones. *

Marca solo un óvalo.

	1	2	3	4	5	
Fácil	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Difícil

Diseño

Valora el diseño de la aplicación.

5. Valoración Personal *

Marca solo un óvalo.

	1	2	3	4	5	
No me gusta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Me encanta

6. Claridad y Limpieza *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

7. Funcionalidad *

Marca solo un óvalo.

	1	2	3	4	5	
Nada funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy funcional

8. ¿Qué le sobra o qué le falta? *

9. Comentarios

Contenido teórico

Valora el contenido del módulo de formación teórico de la aplicación.

10. Exactitud *

Marca solo un óvalo.

	1	2	3	4	5	
Incorrecto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Correcto

11. Claridad *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

12. Concisión *

Marca solo un óvalo.

	1	2	3	4	5	
Muy escueto	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Muy extenso

13. ¿Añadirías algo? *

14. ¿Quitarías algo? *

15. Comentarios

Instrucciones

Valora las instrucciones del juego

16. Claridad *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

17. Utilidad *

Marca solo un óvalo.

	1	2	3	4	5	
No son nada útiles	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Son muy útiles

18. Extensión *

Marca solo un óvalo.

	1	2	3	4	5	
Son demasiado breves	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Son demasiado largas

19. ¿Qué le sobra o qué le falta? *

20. Comentarios

Diseño del
Juego

Valora el diseño del juego y su usabilidad (a partir de lo que has visto y lo que explican las instrucciones)

21. Valoración personal *

Marca solo un óvalo.

	1	2	3	4	5	
No me gusta	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Me gusta mucho

22. Claridad *

Marca solo un óvalo.

	1	2	3	4	5	
No se entiende	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Se entiende perfectamente

23. Funcionalidad *

Marca solo un óvalo.

	1	2	3	4	5	
No parece funcional	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Parece muy funcional

24. ¿Cambiarías algo? ¿Qué? *

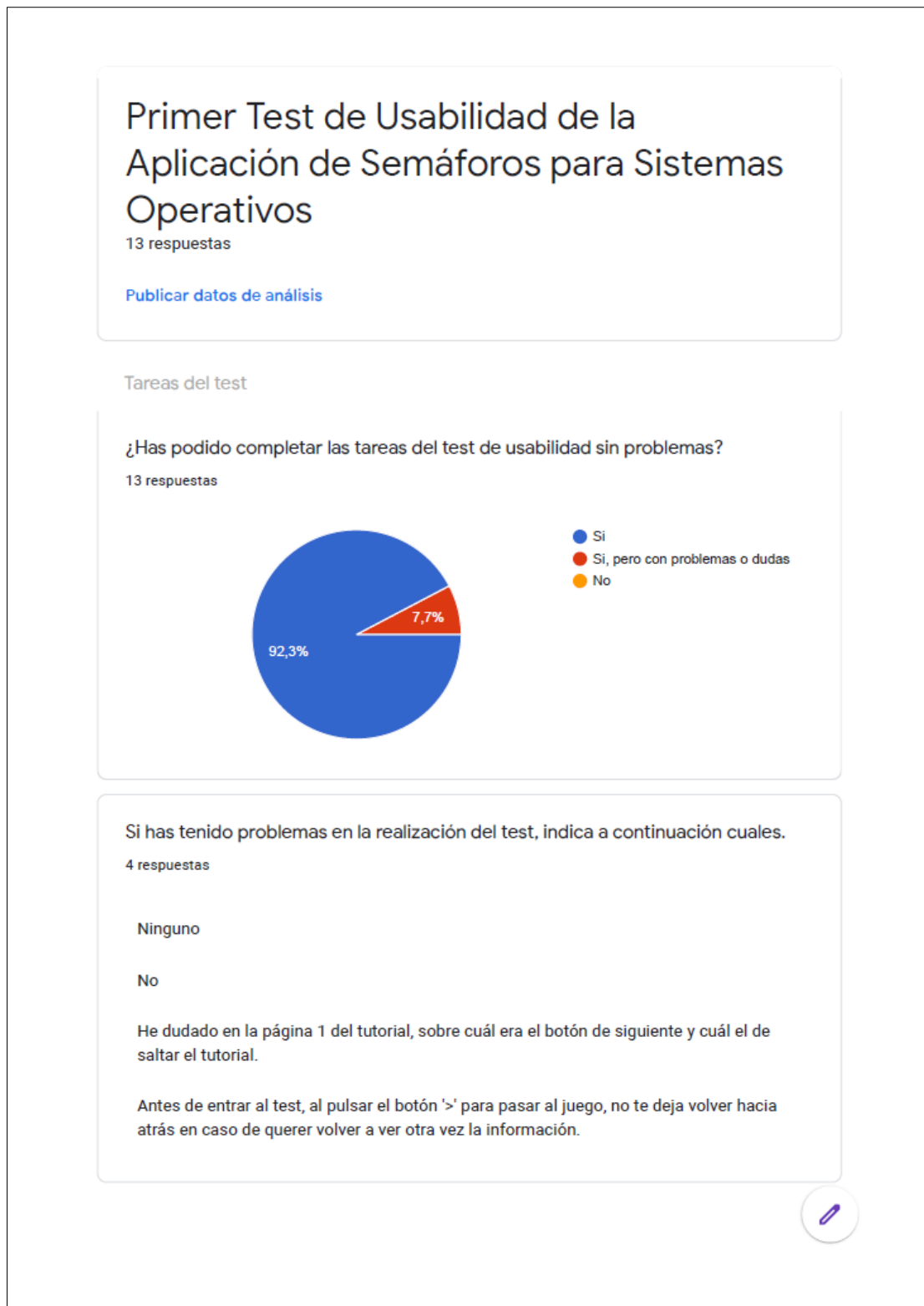
25. Comentarios

Gracias por tu colaboración.

Este contenido no ha sido creado ni aprobado por Google.

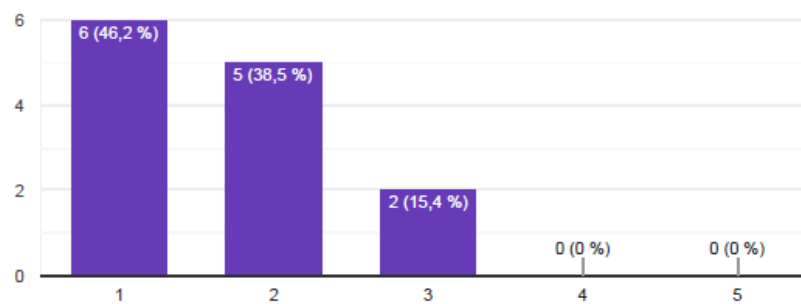
Google Formularios

ANEXO 3: RESPUESTAS AL PRIMER TEST DE USABILIDAD



Valora el nivel de dificultad del test y sus instrucciones.

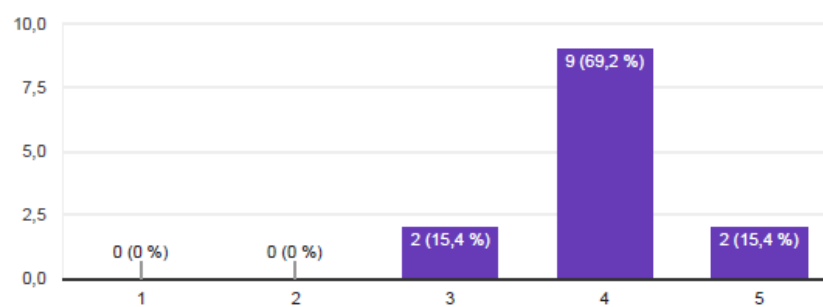
13 respuestas



Diseño

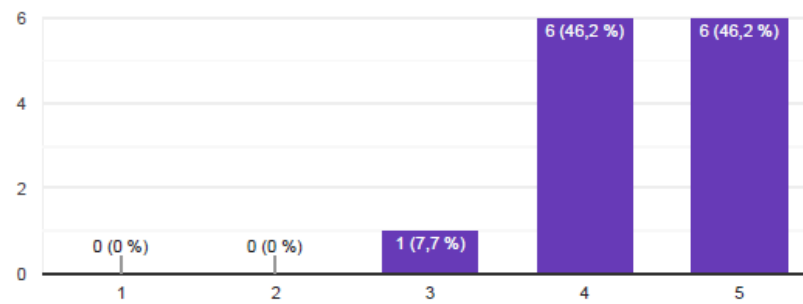
Valoración Personal

13 respuestas



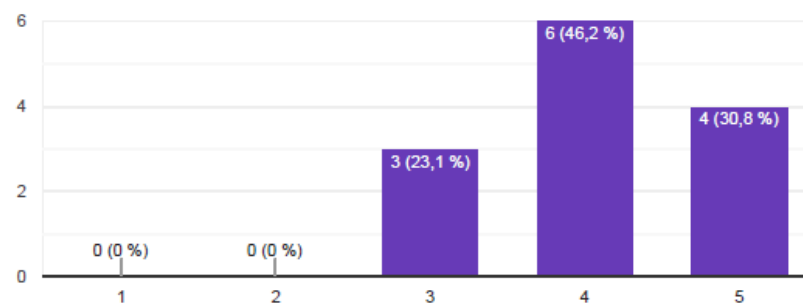
Claridad y Limpieza

13 respuestas



Funcionalidad

13 respuestas



¿Qué le sobra o qué le falta?

13 respuestas

Quizá que las cajas de Semáforo, Acciones, Llamadas y Avisos del Sistema se ajusten para que no quede hueco, y entre todas formen un rectángulo; que los avisos ocupen el ancho de pantalla, y las cajas de encima ocupen todo el hueco.

Como mejora quizá, que la ayuda tenga la opción de ocultar, para el caso de que el usuario repita el proceso y solo se concentre en el funcionamiento.

Modernizar los botones: sin bisel, nada o menos redondeados, colores más simples y llamativos, con letra blanca y efecto al pasar por encima (las clases de Bootstrap siguen bastante la temática).

Cambiar el color de la ventana de avisos del sistema: poner un color diferente al que tiene la ventana de acciones quizás daría una comprensión más visual y directa.

El tutorial es muy claro y simple pero, por decir algo, cuando se pide clicar sobre un botón, además de marcarlo con un icono, sería más visual destacarlo, por ejemplo, difuminando el resto de cosas o rodeándolo con un borde que llame la atención.

En general veo la aplicación muy correcta, cumple sus funciones de manera sencilla,



Comentarios

8 respuestas

Me ha sorprendido gratamente, no me lo esperaba integrado en Moodle. Gran trabajo.

Creo que el diseño de la aplicación es sencillo y claro. Se encuentra muy bien estructurado teniendo a la izquierda dividido entre teoría y juego.

en todo momento tienes muy claro en que parte del tutorial estas.

Esta increíble. Yo voy atrasadísimo.

Entendiendo el rol de usuario como el de un estudiante familiarizado con el entorno Moodle en el que se ha adaptado el trabajo, he visto bastante intuitiva la navegación.

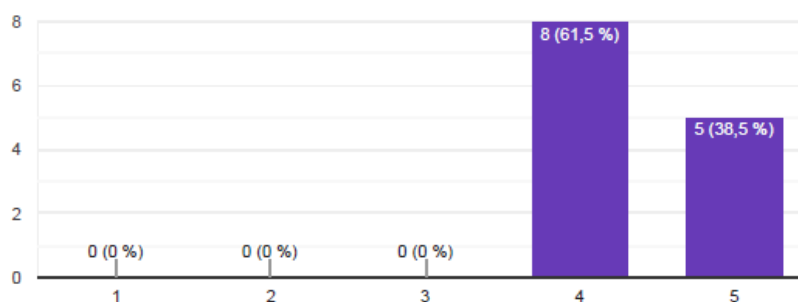
En general tiene un diseño agradable y claro.

Bastante bien en general. He echado en falta un ejemplo muy simple, para que lo entienda alguien que no ha visto nunca lo que es un semáforo. También un mensaje de alerta en caso de pulsar un botón que no tiene nada que ver con lo que estás

Contenido teórico

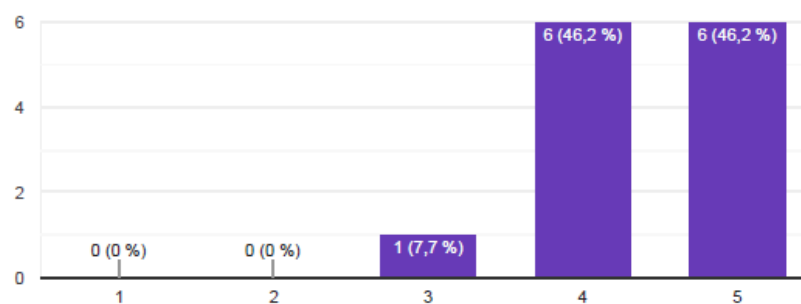
Exactitud

13 respuestas



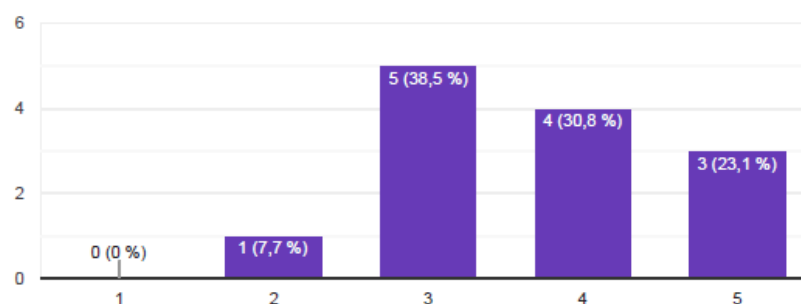
Claridad

13 respuestas



Concisión

13 respuestas



¿Añadirías algo?

13 respuestas

No

Creo que no

Nada

Nada.

Como he comentado anteriormente el tema del ejemplo del taller.

En general las explicaciones son correctas sobre todo para el usuario sin conocimientos de lo que es un semáforo , quizás para los que tenemos conocimientos se hace un poco largo.

no añadiría nada

No. Queda explicado correctamente.

¿Quitarías algo?

13 respuestas

No

Nada, la teoría es correcta y concisa

Nada.

No quitaría nada, puesto que has añadido lo más importante de la teoría. Está muy bien resumido.

Quitaría la parte del código porque para el usuario sin conocimientos no le da ninguna información

no quitaría nada

Está todo lo demás bastante bien, no quitaría nada.

No.



Comentarios

5 respuestas

Creo que hace lo que tiene que hacer y lo describe con precisión

Me ha gustado mucho tu aplicación, creo que es una herramienta que proporciona un mejor entendimiento sobre los semáforos al usuario.

Las instrucciones de la formación teórica me han parecido claras para la navegación entre pantallas, por lo que durante el tutorial del juego no he tenido problemas con el desplazamiento.

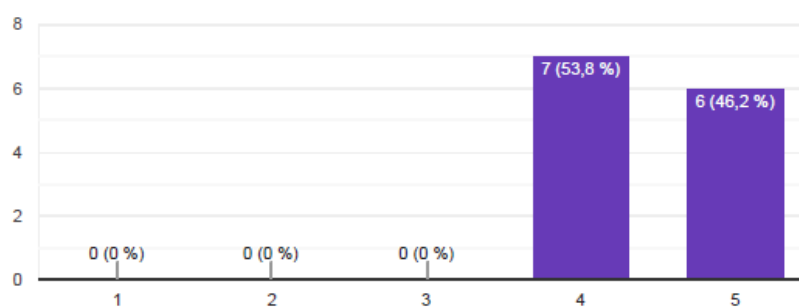
Creo que explica bastante bien el marco teórico que se pretende.

Bastante bien en general.

Tutorial

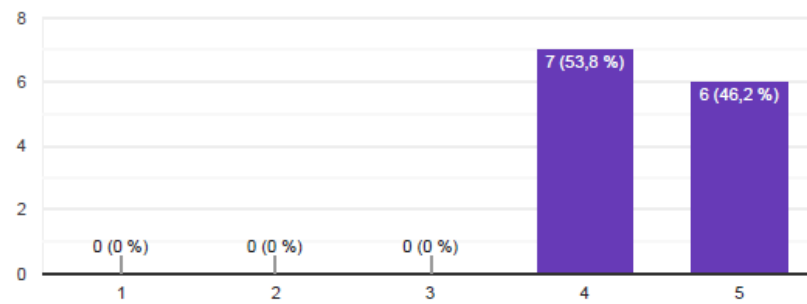
Claridad

13 respuestas



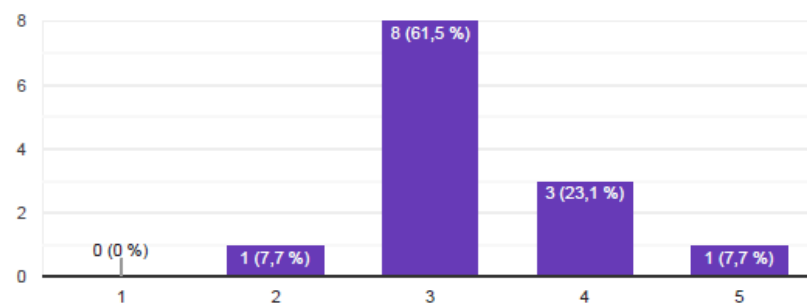
Utilidad

13 respuestas



Extensión

13 respuestas



13 respuestas

En la primera parte del tutorial, iría introduciendo poco a poco las ventanas, aunque eso provocase que el tutorial fuese ligeramente más largo, explicándolas individualmente y luego añadiéndolas al resultado final. Empezando con una sola ventana, seguidamente añadiendo una segunda, etc. Evitando que el alumno pueda saturarse en un primer momento.

Quitaría la opción de "pulsar en siguiente". Puesto que el usuario va a entender perfectamente qué tiene que pulsar en la flecha.

Me ha gustado mucho el tutorial número 4, puesto que te anima a pulsar bloquear proceso y visualmente el usuario aprecia lo que sucede. Igual añadiría en el tutorial otra explicación más interactiva como esta.

No



Comentarios

3 respuestas

En el tutorial, cuando se libera un recurso y se le asigna al proceso 1 no se actualiza el recurso disponible en la caja de semáforo, debería de poner un "1". Opino que cuando apareciese "signal" debería dejar pulsar el botón $s = s + 1$ para que en la caja del semáforo apareciese un recurso disponible, y luego a continuación lo que tú dices de poner el número del proceso al que se le asigna el recurso, como en el ejemplo, una vez que se pone el número del proceso y se le da al botón, desaparece el proceso de la cola como bien pones y luego el contador de semáforo a 0. Yo creo que sería más recomendable que solo dar al botón $s = s + 1$ cuando no haya más procesos en la cola, se haría más visible. El proceso wait lo veo bastante bien explicado aunque pondría algún ejemplo con el contador a más de 0. Por lo demás te felicito, está bastante bien pensado y es muy intuitivo. Me gusta bastante

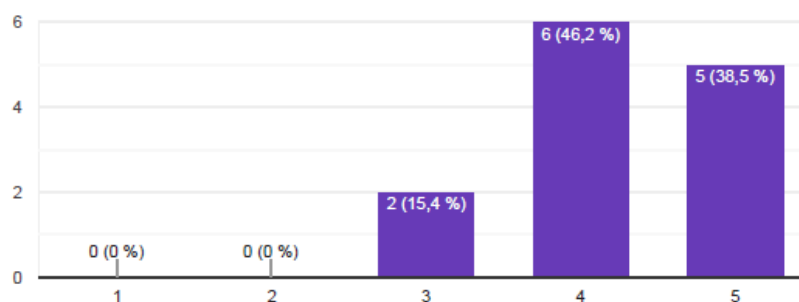
Tiene algunos fallos ortográficos:

- * Página 1. Se escribe Dijkstra, le falta la "k"
- * Página 3. Es "tú", en lugar de "tu", cabría sustituirlo por "el usuario" o "el jugador"
- * Página 8. "... nos avisa de que..", en lugar de "... nos avisa que..". Ya que se avisa de algo, no que algo.

Diseño del Juego

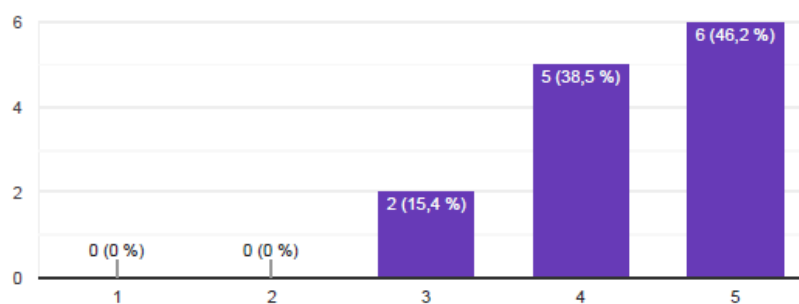
Valoración personal

13 respuestas



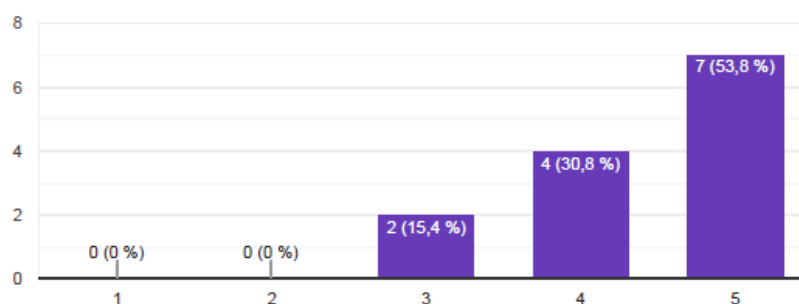
Claridad

13 respuestas



Funcionalidad

13 respuestas



¿Cambiarías algo? ¿Qué?

13 respuestas

Nada más que lo que ya se ha comentado. Quizás no soy de mucha ayuda, pero la idea me ha convencido, quizás el diseño no sea de mi agrado, pero es una forma interesante de aprender y q está bien llevaba a la práctica.

Nada.

Me gusta como está diseñada la aplicación creo que tiene una interfaz clara y sencilla.

Para cambiar algo tendría que probarlo.

El modulo de errores lo pondria mas visible.

no cambiaria nada

Icono de ayuda por si surgen dudas. Se abriría un modal con fragmentos de texto del tutorial en función de donde esté "atrancado".

Comentarios

3 respuestas

En general me ha gustado mucho y creo que es un buen trabajo.

.

En la página 1 del nivel 1, dice: "se cuidadoso", lo correcto sería: "sé cuidadoso".

Gracias por tu colaboración.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios



Primer Test de Usabilidad de la Aplicación de Rendimientos para Sistemas Operativos

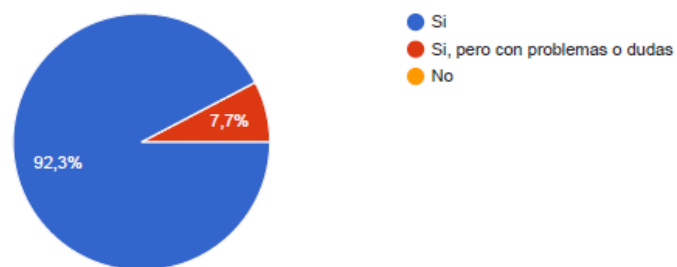
13 respuestas

[Publicar datos de análisis](#)

Tareas del test

¿Has podido completar las tareas del test de usabilidad sin problemas?

13 respuestas



Si has tenido problemas en la realización del test, indica a continuación cuales.

3 respuestas

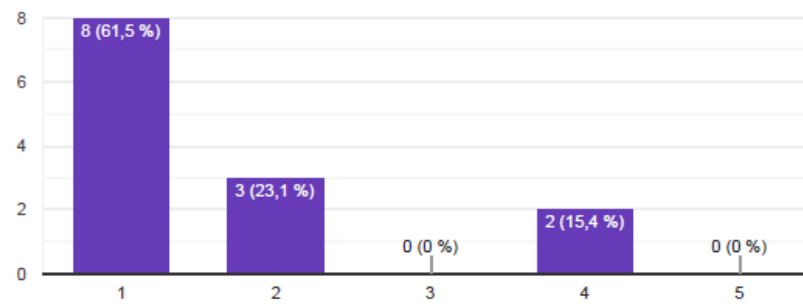
Ninguno

Ninguna



Valora el nivel de dificultad del test y sus instrucciones.

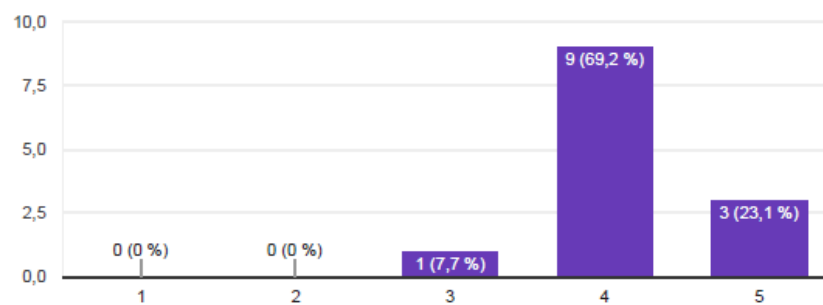
13 respuestas



Diseño

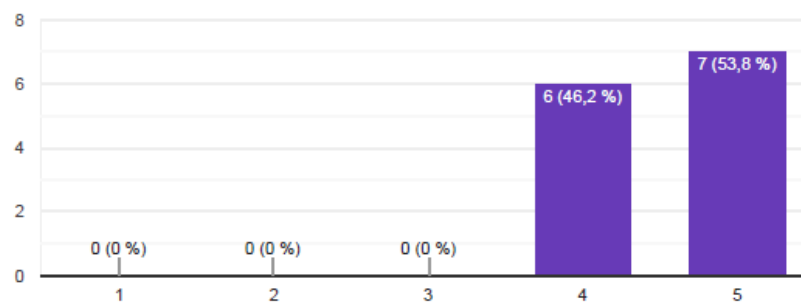
Valoración Personal

13 respuestas



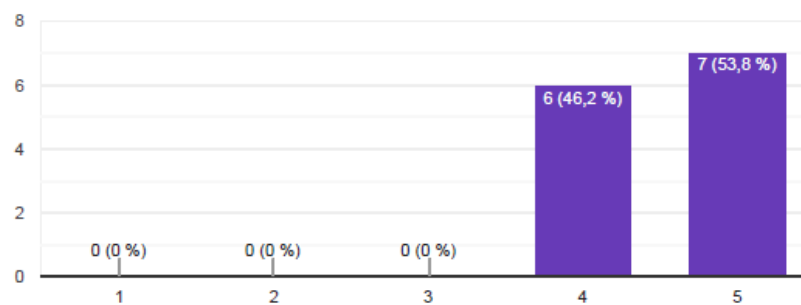
Claridad y Limpieza

13 respuestas



Funcionalidad

13 respuestas



¿Qué le sobra o qué le falta?

13 respuestas

Estaría bien un diseño Moderno, más atractivo visualmente.

En el ejemplo, quizá que destaquen las respuestas sobre la pregunta; para que predomine la sensación de Juego sobre la de Control/Exámen. Creo que daría más sensación de juego.

Cambiaría el punto rojo que indica la página en la que nos encontramos en el menú de la izquierda por algo más moderno y más acorde con el diseño general de la aplicación.

Ns/Nc

Las instrucciones podrian estar más esquemáticas, frases más cortas (tipo esquema)

Nada.

Ni el sobra ni le falta

Comentarios

5 respuestas

Creo que es complicado dar un aspecto de juego a una pregunta del tipo del ejemplo, a no ser que la pregunta se plantee de otra forma

Tiene una interfaz sencilla y agradable.

Bonito diseño

esta muy claro y ademas has destacado los botones con un color que se ve a primera vista

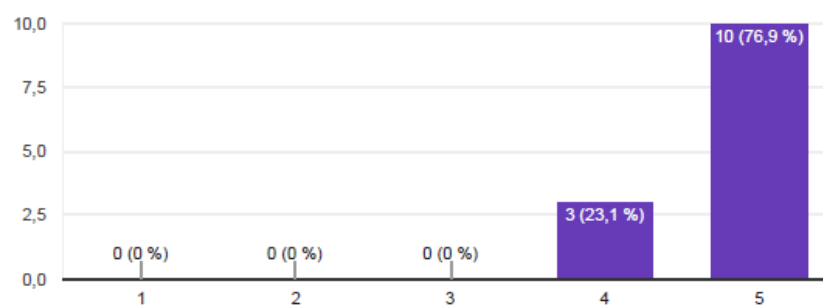
En general es una interfaz clara y sencilla.

Contenido teórico



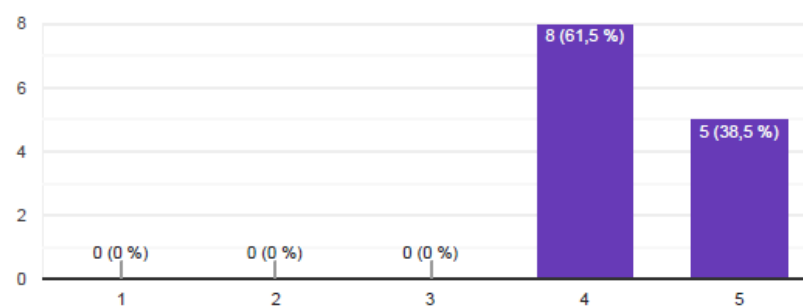
Exactitud

13 respuestas



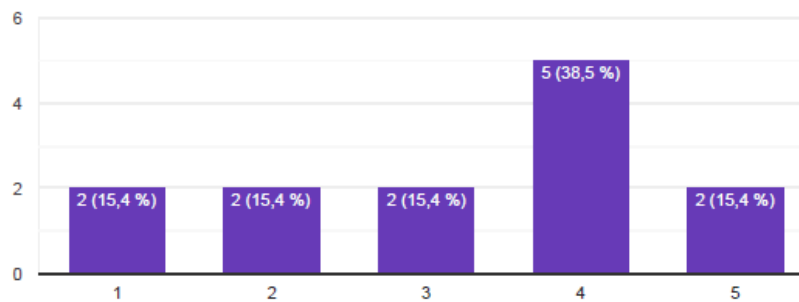
Claridad

13 respuestas



Concisión

13 respuestas



¿Añadirías algo?

13 respuestas

No

Añadiría un ejemplo pasa a paso resolviendo un ejercicio

Añadiría un poco más de marco teórico explicando el tema de rodajas y cambio de contexto, me parece un poco escueto.

No, está todo correcto.

La explicación del RR y del tiempo de ejecución es demasiado escueta, y sin ejemplos que pudiesen ver la usabilidad práctica de esos cálculos. Añadiría algún tipo de ejemplo orientativo para ver mejor la usabilidad de dichos cálculos.

Está todo correcto

jugabilidad real

no



¿Quitarías algo?

13 respuestas

No

Nada

Dividiría la pregunta en 3 para hacer los ejercicios más claros e incluso más sencillos de adaptar a personas con problemas visuales.

No.

La formacion teorica está adecuada

Nada.

no

En la parte visual quitaría (pagina 1) puesto que el usuario fijándose en la izquierda en el punto rojo sabe perfectamente que se encuentra en en dicho módulo.

Comentarios

4 respuestas

El contenido de las preguntas es correcto, nada que objetar al respecto

Bastante bien en general

Me ha gustado el sistema de fichas explicando conceptos distintos en cada página y simplificando la explicación de como realizar los cálculos. Así es más fácil retener la información.

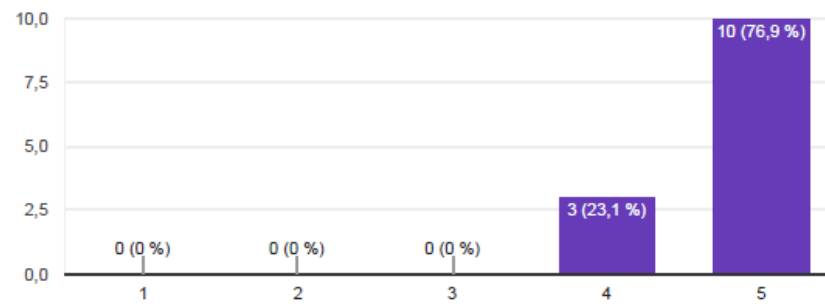
es fácil de leer y se entiende muy bien

Instrucciones



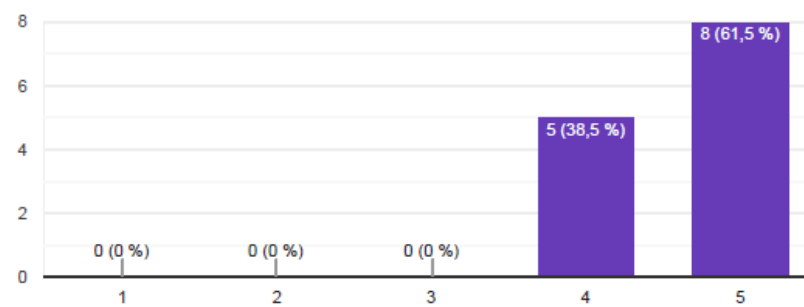
Claridad

13 respuestas



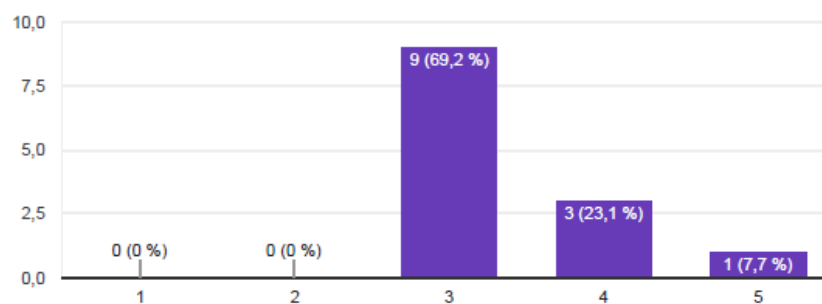
Utilidad

13 respuestas



Extensión

13 respuestas



¿Qué le sobra o qué le falta?

13 respuestas

Nada, clara y concisa

Quizá le falte instrucciones "desplegables" si fuera necesario en algún tipo de respuesta del tipo "arrastrar valores a una caja" (si se diera el caso)

Está muy correcto.

Nada

Quizás además de después antes de iniciar el juego pondría las puntuaciones actuales para motivar al lector.

Nada.

No le sobra nada

Quiero jugar cuando acabes el desarrollo



Comentarios

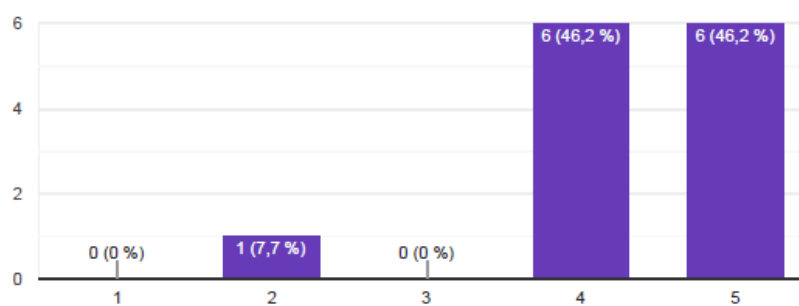
1 respuesta

Está todo perfectamente explicado

Diseño del Juego

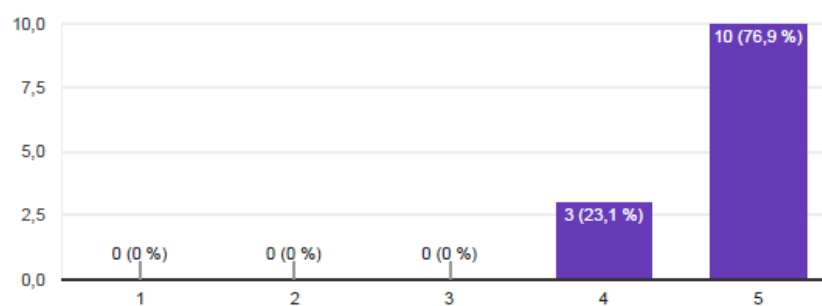
Valoración personal

13 respuestas



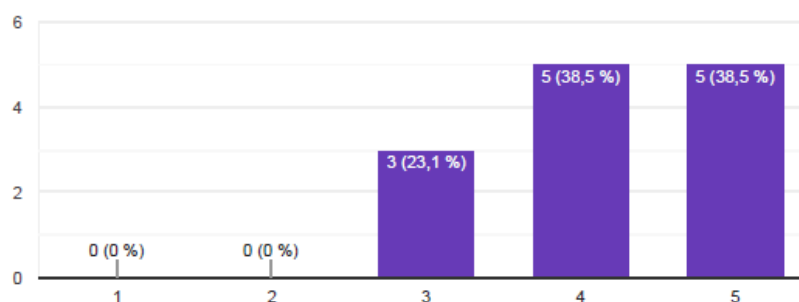
Claridad

13 respuestas



Funcionalidad

13 respuestas



¿Cambiarías algo? ¿Qué?

13 respuestas

No

Son preguntas tipo test, me recuerda a las típicas unidades de inglés con diapositivas y preguntas a rellenar al final, después del juego de los semáforos, esperaba algo más interactivo e interesante. En este caso, entiendo que es más difícil, pero creo que habría que darle una vuelta de tuerca más

Preguntas lo más simples posibles, como comentaba, el ejemplo dividirlo en 3 preguntas para simplificarlo y poder disponer de más pantalla por si quisiera usar otro tipo de componentes para la respuesta.

El diseño de la tabla de preguntas y respuestas creo que es mejorable, se podría poner algo más acorde con el resto del diseño de la aplicación, por ejemplo en la paleta de colores utilizada.

Para ver su funcionalidad habría que probarlo real.

Todo esta muy claro



Comentarios

2 respuestas

Pero en general, el objetivo buscado creo que está logrado. Mis aportaciones son sugerencias. Sigo diciendo lo mismo, me encanta la idea de integrar el juego en Moodle. Gran trabajo.

En mi humilde opinión añadiría una calculadora de "tiempo de ejecución de los procesos", es decir, un botón con el valor de la rodaja de tiempo y otro con el valor de tiempo entre rodajas, y un marcador con los tiempos de los tres procesos, donde cada vez que pulsas un botón de rodaja de tiempo o de valor de tiempo entre rodajas se van decrementando los valores hasta que en los tres esté a 0. Así se sabría la idea de a qué botón pulsar y cuantas veces obteniendo el resultado final en otra caja con la acumulación de los tiempos de ejecución de todos los procesos.

Gracias por tu colaboración.

Este contenido no ha sido creado ni aprobado por Google. [Notificar uso inadecuado](#) - [Términos del Servicio](#) - [Política de Privacidad](#)

Google Formularios



ANEXO 4: DETALLES DEL CÓDIGO DE LA PLATAFORMA

Control de enrutamiento de CodeIgniter:

```
/**
 * -----
 * Route Definitions
 * -----
 */

// We get a performance increase by specifying the default
// route since we don't have to scan directories.
// Si hago control de apps, redirigir el home alli.
$routees->get('apps/(:any)', 'Aplicaciones::control/$1');
$routees->get('apps', 'Aplicaciones::index');
$routees->match(['post'], 'apps/ajaxRequestPost', 'Aplicaciones::ajaxRequestPost');
$routees->get('(:any)', 'Pages::showme/$1');
```

Funciones del controlador de aplicaciones y detalle de la función ajaxRequestPost:

```
<?php namespace App\Controllers;
use App\Models\AppsModel;
use App\Models\MenuModel;
use App\Models\SessionModel;
use App\Models\ProgressModel;
use App\Models\ScoreModel;
use App\Models\MaxscoreModel;
use App\Models\ContentModel;
use CodeIgniter\Controller;

class Aplicaciones extends Controller
{
    public function index()
    {

    }

    public function control($aplicacion = NULL, $clase = NULL, $pagina = NULL, $usertemp = NULL)
    {

    }

    public function show($aplicacion, $clase = NULL, $pagina = NULL)
    {

    }

    public function gestUser()
    {

    }

    public function ajaxRequestPost() {

        $modelscore = new ScoreModel();

        //Leer datos de puntuacion e insertar en BBDD
        $datas = array(
            'idUser' => $this->request->getVar('idUser'),
            'idApp' => $this->request->getVar('idApp'),
            'fecha' => $this->request->getVar('fecha'),
            'puntuacion' => $this->request->getVar('puntuacion'),
            'tiempo' => $this->request->getVar('tiempo')
        );
        $modelscore->insert($datas);

    }
}
```

Modelo de datos de Aplicaciones:

```
<?php namespace App\Models;

use CodeIgniter\Model;

class AppsModel extends Model
{
    protected $table = 'aplicaciones';

    public function getApps($slug = false)
    {
        if ($slug === false)
        {
            return $this->findAll();
        }

        return $this->asArray()
            ->where(['slug' => $slug])
            ->first();
    }
}
```

Modelo de datos de menús:

```
<?php namespace App\Models;

use CodeIgniter\Model;

class MenuModel extends Model
{
    protected $table = 'menus';
    protected $primaryKey = 'idApp';

    public function getMenu($idApp = false)
    {
        if ($idApp === false)
        {
            return $this->findAll();
        }

        return $this->where('idApp', $idApp)
            ->findAll();
    }
}
```

Detalle del generador de problemas y soluciones (parte de la función randomQ) del juego de rendimientos:

```
function randomQ () {
  // Siguiendo problema
  window.level = window.level + 1;

  // Problema aleatorio:
  // 1 = utilizacion, 2 = rendimiento
  tipo = Math.floor(Math.random() * 2) + 1;

  // En función del tipo de problema generamos los datos aleatorios
  if (tipo == 1) {
    // Problema de porcentaje de utilización de CPU
    // Generamos los datos aleatorios
    // Instrucciones maquina por microsegundo (10-100) step 5
    instcpu = 5*(Math.floor(Math.random() * 21) + 2);

    // Tiempo lectura (10-20)
    tlec = Math.floor(Math.random() * 11) + 10;

    // Tiempo escritura (10-20)
    tesc = Math.floor(Math.random() * 11) + 10;

    // Instrucciones maquina proceso (50-200) step 5
    instproc = 5*(Math.floor(Math.random() * 31) + 10);

    // Respuesta correcta
    respcorr = Math.round((instproc/instcpu)/(tlec + tesc + (instproc/instcpu))*10000)/100;

    // Respuestas falsas
    respmal = ["", ""];
    respmal[0] = Math.round((instproc/instcpu)/(tlec + (2*(instproc/instcpu)))*10000)/100;
    respmal[1] = Math.round((instproc/instcpu)/(tesc + tesc + (instproc/instcpu))*10000)/100;

    if (respmal[0] == respcorr) {
      respmal[0] = respmal[0] + 1;
    }

    if (respmal[1] == respcorr) {
      respmal[1] = respmal[1] + 1;
    }
  }
}
```

Detalle del cuerpo (appbody.php) del gestor de aplicaciones:

```
<!-- Cuerpo -->
<div id="nav-mainpage" data-region="mainpage" tabindex="0">
  <div class="maincontent">
    <?php

    if (!empty($pagina['pExterna']))
    {
      //Se carga la pagina externa
      include $pagina['pExterna'];
    } else {
      // Se carga el contenido de BBDD
      echo $pagina['contenido'];
    }
    ?>

    <?php if ($pagina['pie']=='1') : ?>
    <nav class="nav-body-foot">
      <div class="img-ant">
        <?php if (!empty($pagina['anterior'])) : ?>
        <a href="<?="/apps" . $pagina['anterior']?>"></a>
        <?php else : ?>
        
        <?php endif ?>
      </div>
      <div class="img-saltar">
        <?php if (!empty($pagina['saltar'])) : ?>
        <a href="<?="/apps" . $pagina['saltar']?>"></a>
        <?php endif ?>
      </div>
      <div class="img-sig">
        <?php if (!empty($pagina['siguiente'])) : ?>
        <a href="<?="/apps" . $pagina['siguiente']?>"></a>
        <?php else : ?>
        
        <?php endif ?>
      </div>
    </nav>
    <?php endif ?>

  </div>
</div>
```

ANEXO 5: SEGUNDO TEST DE USABILIDAD

Instrucciones del Segundo Test de Usabilidad de las Aplicaciones de Semáforos y Rendimientos para la asignatura de Sistemas Operativos.

Gracias por tu ayuda en la evaluación de los prototipos de las aplicaciones de apoyo a la formación de la asignatura de Sistemas Operativos.

En el Test se desea comprobar la usabilidad de la implementación en la plataforma final de las aplicaciones y, sobre todo, la jugabilidad de las aplicaciones formativas. Dado que se han implementado mejoras y cambios de acuerdo a los resultados del primer test, deberemos repasar los apartados ya revisados en el primer test pero nos centraremos en el apartado de los juegos que no se incluía en el primer test.

Ten en cuenta que este test no se realiza sobre una maqueta gráfica sino sobre la plataforma final, aunque aún se encuentre en desarrollo. Para simplificar la operativa, se han activado previamente todas las funciones que se activan al completar ciertas acciones.

Una vez hayas completado las tareas te haré una pequeña batería de preguntas para valorar tus sensaciones sobre el entorno y el test. Si prefieres contestarlas con calma puedo enviarte la encuesta y me la envías una vez completada.

Para la realización del test debes suponer que estas accediendo a la formación online, a través de la plataforma Moodle de la Udima, de los temas de semáforos y rendimientos. Las tareas que has de realizar son:

- 1- Usa el enlace a continuación para acceder a la formación online:

<https://ssoo.anraingenieros.es/juegosSSOO.php>

- 2- Accede al módulo de semáforos mediante el enlace adecuado.
- 3- Lee la información de la página de introducción y completa el recorrido por el módulo de formación teórica
- 4- Usando los controles, accede a la página de instrucciones del juego y completa el tutorial del mismo. No te recomiendo que lo saltes, aunque ahora puedes hacerlo, puesto que ha sido rediseñado desde la versión anterior.
- 5- Accede ahora, al juego y disfruta de una partida.
- 6- Comprueba tu puntuación. ¿Crees que puedes mejorarla?
- 7- Vuelve a la página inicial y accede ahora al módulo de Rendimientos.
- 8- Lee la página de introducción, completa el recorrido teórico y accede a la página de instrucciones.

- 9- Disfruta de una partida y comprueba tus resultados.
10- Completa la breve encuesta y dame tus valoraciones.

Espero que la experiencia haya sido satisfactoria y te agradezco enormemente tu colaboración.

Muchas gracias.

Javier José García García

Preguntas

- 1- ¿La implementación en la plataforma final ha mejorado o empeorado la impresión que tenías desde el primer prototipo?
 - a. Empeorado mucho
 - b. Empeorado
 - c. Indiferente
 - d. Mejorado
 - e. Mejorado Mucho
- 2- ¿El rediseño de los juegos respecto al esbozado en el primer prototipo?
 - a. No me ha gustado nada
 - b. No me ha gustado
 - c. Indiferente
 - d. Me ha gustado un poco
 - e. Me ha gustado mucho
- 3- La jugabilidad de los juegos es:
 - a. Escasa
 - b. Regular
 - c. Indiferente
 - d. Alta
 - e. Muy alta
- 4- La interacción con la plataforma y los juegos es:
 - a. Muy mala
 - b. Mala
 - c. Regular
 - d. Buena
 - e. Muy buena
- 5- ¿Crees que si no supieras nada de Semaforos y rendimientos habrías aprendido algo con las plataformas?
 - a. Nada
 - b. Poco
 - c. Algo
 - d. Bastante
 - e. Mucho
- 6- ¿Crees que estas aplicaciones resultarían útiles para el aprendizaje en el aula virtual de la UDIMA?
 - a. No
 - b. Si
- 7- Haz aquellos comentarios y aportaciones que creas interesantes para la mejora de la plataforma.